P. 1877/83

10 1983

informatyka

SIMBOL
Systemy pomiarowe
FORTRAN przemysłowy
Języki symulacyjne
PROSTR
Historia SZBD

WYDAWNICTWO SIGMA

Do autorów

Nie publikujemy precyzyjnych zaleceń dla autorów, by nie peszyć ich natłokiem dyrektyw i ograniczeń. Podstawowe zasady, jakimi kierujemy się przy kwalifikacji artykułu i warunki formalne, jakie powinien spełniać maszynopis, chcemy jednak co jakiś czas przypominać.

Wiele zasad oczywistych jest już przy pobieżnym przejrzeniu pisma: jak dzielić artykuł na części, co wyróżniać, w jakiej formie umieszczać bibliografię, jak zapisywać własny życiorys. Natomiast na elementarne pytanie — po co piszę — odpowiedzieć sobie musi przede wszystkim sam autor.

Podstawowym kryterium merytorycznym materiału (nie licząc głównego kryterium — oceny z punktu widzenia dotychczasowej wiedzy informatycznej) jest społeczna użyteczność zawartych w tekście rozważań i informacji. To w końcu oczywiste — jesteśmy powołani przede wszystkim do tego, aby służyć czytelnikom. W pierwszej kolejności liczyć się musi zatem interes społeczny. O tym winni także pamiętać autorzy.

Objętość tekstu powinna być wyznaczona przez społeczną wagę tematu, jego konstrukcja — przez wymóg czytelnego przekazu, zaś forma — przez dostosowanie języka i sposobu argumentacji do możliwości percepcyjnych przeciętnego odbiorcy. To truizmy, niemniej — jak praktyka wskazuje — warte przypomnienia. Autorzy najczęściej nie zadają sobie trudu, by wprowadzić czytelnika w zawiłości swoich specjalistycznych wywodów, nie biorą pod uwagę miejsca publikacji artykułu, tzn. charakteru naszego czasopisma.

Jeśli zaś chodzi o wymogi formalne, nie są duże. Co do objętości — maszynopis nie powinien przekraczać 12 znormalizowanych stron (30 wierszy z 60 znakami w wierszu). Jeśli jednak interesujący, zgodny z profilem pisma, temat wymaga większej objętości, to oczywiście względy formalne nie stoją na przeszkodzie. Rysunki techniczne, dołączane do tekstu, są w redakcji przygotowywane przez wykwalifikowanego kreślarza, nie muszą być zatem dostarczane na kalce; ważne natomiast, aby były całkowicie czytelne, przejrzyste i zachowujące konieczne proporcje.

Pożądany jest również dodatkowy materiał ilustracyjny (kontrastowe zdjęcia czarno-białe, wydruki, itp.), jeśli wzbogaca on bądź precyzuje wywód autora.

Maszynopis powinien być nadsyłany w dwóch egzemplarzach (oryginał plus kopia), z załączeniem aktualnych adresów, a także numerów telefonów, umożliwiających szybki kontakt. Autorów artykułów problemowych, którzy nie zamieszczali swych prac w ciągu dwóch ostatnich lat, prosimy ponadto o załączenie zdjęcia oraz życiorysu.

Autorzy, którzy chcieliby nawiązać bliższy kontakt z Czytelnikami, proszeni są o zamieszczenie adnotacji, iż godzą się na podanie na naszych łamach swojego adresu bądź telefonu.

REDAKCJA

NACZELNA ORGANIZACJA TECHNICZNA
WYDAWNICTWO
SIGMA
CZASOPISM I KSIĄŻEK TECHNICZNYCH
ul. Świętokrzyska 14a
00-950 Warszawa
skrytka pocztowa 1004

KOLEGIUM REDAKCYJNE

Redaktor naczelny: prof. dr hab. Leon ŁUKASZEWICZ

mgr inż. Zbigniew GLUZA, dr Janusz GWIAZDA, Władysław KLEPACZ (zastępca redaktora naczelnego), dr inż. Tomasz PAWLAK, mgr Marek SOBCZYK, mgr Andrzej SZAŁAS, mgr Zbigniew ŚWIRSKI, dr inż. Janusz ZALEWSKI

Sekretarz redakcji: mgr Teresa JABŁOŃSKA

RADA PROGRAMOWA

Prof. dr hab. Tadeusz PECHE (przewodniczący), mgr inż. Tomasz BAŃKOWSKI (sekretarz), mgr inż. Antoni BOSSOWSKI, mgr inż. Roman BURNO, prof. dr hab. Andrzej JANICKI, mgr inż. Jan KRAMARCZUK, prof. dr hab. inż. Juliusz KULIKOWSKI, prof. dr hab. Leon ŁUKASZEWICZ, prof. dr hab. Antoni MAZURKIEWICZ, gen. dr inż. Marian PASTERNAK, dr inż. Bronisław PIWOWAR, mgr Zbigniew SUBSTYK, prof. dr hab. Tadeusz WALCZAK

Materiałów nie zamówionych Redakcja nie zwraca.

Redakcja: 00-041 Warszawa, ul. Jasna 14/16, pok. 243 i 244, tel. 27-71-40, dyżury redakcji 10.00—12.00

Zakł. Graf. „Tamka". Zam. 2257. Obj. 5,0 ark. druk. Nakład 3550 egz. M-95.

Cena egzemplarza zł 75.— INDEKS 36124 Prenumerata roczna zł 900.—

P. 1877/83

Informatyka

zastosowania w gospodarce, technice i nauce

Nr 10
MIESIĘCZNIK
1983
ROK XVIII
Październik

ORGAN KOMITETU INFORMATYKI, MINISTERSTWA NAUKI, SZKOLNICTWA WYŻSZEGO I TECHNIKI ORAZ KOMITETU NAUKOWO-TECHNICZNEGO NOT DS. INFORMATYKI

W NUMERZE: **Strona**

Język SIMBOL dla MERY 400
Jerzy T. Majewski, Krzysztof Wagner 2

Programowane systemy pomiarowe
Elżbieta Jezierska-Ziemkiewicz, Jan Ryżko 5

Użycie mikroprocesorów 16-bitowych w przetwarzaniu rozłożonym
Janusz Zalewski 8

Przemysłowy FORTRAN czasu rzeczywistego. Część 1
Tłum. Kazimierz Maliszewski 11

Aspekty użytkowe języków symulacyjnych (2). Symulacja ciągła i ciągło-dyskretna
Teresa Wilczek 15

SYSTEMY

Pakiet dokumentowania i wspomagania prac programistycznych PROSTR
Zygmunt Ryznar 18

HISTORIA

Historia Systemów Zarządzania Bazami Danych
Oprac. Grażyna Siara 21

Z KRAJU

Zniszczyć CYFRONET?
Zbigniew Gluza 24

Seminarium minikomputerowe
Anna i Witold Lucińscy 26

Historia warszawskiego ETOBU (1953—1983)
Kazimierz Pakulski 28

Poczucie gruntu (rozmowa z Marianem Urazem, dyrektorem warszawskiego ETOBU)
Oprac. Z. Gluza 30

GIEŁDA INFORMACJI 24—25

ZE ŚWIATA

Komputery i urządzenia peryferyjne
Oprac. Jacek Walas 31

Firmy amerykańskie
Oprac. Jan Ryżko 34

Kierunki rozwoju oprogramowania w fizyce wielkich energii
Piotr Strzałkowski 35

TERMINOLOGIA

Terminologia FORTRANU czasu rzeczywistego
Kazimierz Maliszewski 37

W SKRÓCIE 38

POGLĄDY

Komputery zagrażają!
Janusz Gwiazda III okł.

JERZY T. MAJEWSKI
KRZYSZTOF WAGNER
Zakłady Przyrządów Pomiarowych i Systemów Minikomputerowych
Warszawa

Język SIMBOL dla MERY 400

W Polsce pracuje obecnie ok. 600 konfiguracji minikomputera MERA 400, a produkcja jest kontynuowana. W oprogramowaniu MERY 400 brakowało dotychczas języka programowania zorientowanego na przetwarzanie danych. W drugiej połowie 1982 r. autorzy poniższego artykułu zakończyli prace nad kompilatorem języka SIMBOL, który wypełnił tę lukę. Pod koniec ubiegłego roku rozpoczęto sprzedaż kompilatora wraz z pełną dokumentacją, zawierającą również podręcznik programowania w języku SIMBOL. W eksploatacji i sprzedaży znajduje się również pierwszy system użytkowy zaprogramowany w języku SIMBOL — System Ewidencji Środków Trwałych *EST*. Z pierwszych doświadczeń użytkowników wynika, że język SIMBOL zaspokaja postawione przed nim wymagania.

SIMBOL (SIMple Business Oriented Language) jest problemowym językiem programowania, przeznaczonym głównie do przetwarzania danych, choć nie wyklucza to możliwości jego użycia również w innych dziedzinach zastosowań. Przystępując do projektowania tego języka założono, że powinien on:

- nakłaniać do dobrego stylu programowania, umożliwiając klarowny zapis algorytmu oraz podejście zwane programowaniem strukturalnym
- zawierać rozbudowany aparat operowania zbiorami danych o bogatej strukturze i różnorodnych sposobach dostępu
- umożliwiać wygodne posługiwanie się operacjami wejścia-wyjścia wraz z kontrolą danych wejściowych i łatwym redagowaniem wydruków
- umożliwiać implementację na małym komputerze o ograniczonej pamięci operacyjnej.

BUDOWA PROGRAMU

Programy w języku SIMBOL składają się z segmentów stanowiących wyodrębnione fragmenty algorytmu. Każdy segment składa się z części zawierającej deklaracje obiektów oraz z instrukcji złożonej opisującej wszystkie czynności danego segmentu. Istnieją trzy rodzaje segmentów:
PROGRAM — główny segment rezydującej części programu; jego nazwa stanowi jednocześnie nazwę całego programu

OVERLAY — główny segment nakładki; jego nazwa stanowi jednocześnie nazwę całej nakładki

PROCEDURE — wyodrębniony w postaci segmentu podprogram dołączany do części rezydującej lub do nakładki; może zawierać parametry formalne.

DANE

Dane (stałe i zmienne) mogą przyjmować wartości różnych typów (podane ograniczenia dotyczą implementacji na minikomputerze MERA 400), a mianowicie:

INTEGER — jest zbiorem liczb całkowitych, z przedziału —32767, +32767

REAL — jest podzbiorem liczb rzeczywistych, którego elementy mają w odniesieniu do modułu wartości nie większe niż $0{,}17 \cdot 10^{39}$ i dokładność równą 11 cyfrom znaczącym

DOUBLE — różni się od typu REAL jedynie dokładnością (23 cyfry znaczące)

TEXT — ciągi znakowe o różnej długości

LOGICAL — ma dwie wartości logiczne: „prawda" i „fałsz",
a ponadto:

CAN — typ specyficzny dla implementacji na MERZE 400, zawierający ciągi znakowe z ograniczonego repertuaru znaków i oszczędniejszej niż typ TEXT reprezentacji maszynowej.

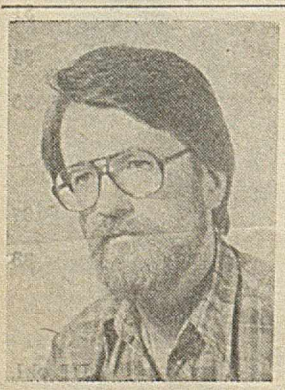
Zmienne w SIMBOLU występują w postaci zmiennych pojedynczych oraz jednowymiarowych tablic. Każda zmienna musi być zadeklarowana.

Przykład:

```
SINGLE  INTEGER (I, K=1), REAL (CENA = 3.55,
                                WARTOŚĆ),
        TEXT (NAZWA 18);
ARRAY   REAL (X (100), Y (256)),
        TEXT (SYMBDOK 2(4)='PZ'/'RW'/'MM'/'WZ')
```

W powyższym przykładzie zmiennej całkowitej K, zmiennej rzeczywistej CENA i wszystkim elementom tablicy tekstowej SYMBDOK nadano wartości początkowe.

Wartość początkowa zmiennych nie inicjowanych explicite jest równa 0 — dla typów liczbowych, spacje — dla typów tekstowych i „fałsz" dla typu logicznego.



Mgr JERZY T. MAJEWSKI jest absolwentem SGPiS (1972 — Organizacja Przetwarzania Danych). Pracował w IMM, OBRTKiP, a obecnie w Zakładach Wytwórczych Przyrządów Pomiarowych i Systemów Minikomputerowych. Zajmuje się oprogramowaniem minikomputerów MERA, m.in. w zakresie organizacji zbiorów, translatorów i systemów użytkowych.

Mgr KRZYSZTOF WAGNER jest absolwentem SGPiS (1973 — Organizacja Przetwarzania Danych). Pracował w IMM, OBRTKiP, a obecnie w Zakładach Przyrządów Pomiarowych i Systemów Minikomputerowych. Zajmuje się oprogramowaniem podstawowym i użytkowym minikomputera MERA 400.

WYRAŻENIA

Składnia wyrażenia oraz reguły obliczania jego wartości (wartościowania) są analogiczne do przyjętych w matematyce. Kolejność oraz reguły wartościowania wyrażenia są następujące:

— obliczanie części wyrażeń w nawiasach
— wykonywanie operacji o wyższym priorytecie przed operacjami o niższym priorytecie
— w pozostałych przypadkach wartościowanie dokonywane jest od prawej do lewej strony wyrażenia.

Operatory występujące w wyrażeniach

Operator	Rodzaj operacji	Typ argumentów		Typ wyniku
*, /	arytmetyczny	INTEGER, REAL, DOUBLE	a)	zgodny z typem argumentu o większej dokładności
+ unarny, — unarny, +, —				
>, <, =, <=, >=, <>	relacja	INTEGER, REAL, DOUBLE	a)	LOGICAL
		TEXT, CAN, LOGICAL	b)	
NOT, AND, OR	logiczny	LOGICAL		LOGICAL

a) — dozwolone mieszanie typów argumentów
b) — argumenty muszą być tego samego typu

Tabela charakteryzuje operacje, jakie mogą wystąpić w wyrażeniu. Operatory zostały wymienione w kolejności od najwyższego do najniższego priorytetu, z tym że operatory umieszczone w jednym wierszu mają równy priorytet.

Wyrażenia występują w instrukcjach podstawienia lub jako warunki logiczne w instrukcjach strukturalnych.

PRZEGLĄD INSTRUKCJI

Instrukcje strukturalne w języku SIMBOL pozwalają w klarowny sposób odzwierciedlić budowę algorytmu.

Instrukcja podstawienia nadaje zmiennej wartość innej zmiennej, stałej lub wyrażenia. W przypadku niezgodności typów po obu stronach symbolu podstawienia (:=) dokonana jest odpowiednia konwersja.

Konwersje danych pomiędzy postacią binarną a znakową, kontrolę postaci danych oraz ich redagowanie zapewniają **instrukcje redagujące**, a także **instrukcje wejścia-wyjścia**, które ponadto powodują przesłanie danych w postaci znakowej.

Instrukcje obsługi zbiorów dokonują odczytu i zapisu rekordów oraz kilku innych operacji na zbiorach.

Istnieje ponadto grupa **instrukcji sterujących:**

GOTO <etykieta> — **instrukcja skoku**
EXECUTE <podprogram> — **instrukcja wykonania pod-**
CALL <procedura> [(<lista parametrów>)] — **instrukcja wykonania procedury**
LOAD <nakładka> — **instrukcja załadowania i wykonania nakładki**
RETURN — **instrukcja wyjścia z segmentu**
PAUSE [<komunikat>] — **instrukcja zawieszenia programu**
STOP [<komunikat>] — **instrukcja zakończenia działania programu.**

INSTRUKCJE STRUKTURALNE

Instrukcja złożona składa się z sekwencji instrukcji rozdzielonych średnikami i ujętych w nawiasy składniowe BEGIN i END.

Istnieją dwie instrukcje warunkowe: IF i CASE. Instrukcja IF występuje w dwóch wersjach, często spotykanych w językach programowania:

```
IF <warunek> THEN <instrukcja 1> ELSE
                                  <instrukcja 2>
```

oraz

```
IF <warunek> THEN <instrukcja>
```

Instrukcja CASE ma następującą postać:

```
CASE <zmienna> OF
        <lista stałych> : <instrukcja 1>;
        <lista stałych> : <instrukcja 2>;
                .
                .
                .
        [OTHERWISE <instrukcja n>]
END
```

Wykonywana jest ta instrukcja wewnętrzna, przed którą umieszczona jest stała równa wartości <zmiennej>. Jeżeli wartość zmiennej nie występuje w żadnej z <list stałych> wewnątrz instrukcji CASE, to wykonywana jest instrukcja poprzedzona słowem OTHERWISE, natomast gdy taki zwrot nie występuje — nie jest wykonywana żadna czynność.

SIMBOL zawiera trzy **instrukcje iteracyjne:** FOR, WHILE oraz REPEAT.

Instrukcja FOR o postaci:

```
FOR <zmienna sterująca>:=<początek>, <granica>
                        [,<krok>] DO <instrukcja>
```

powoduje powtórzenie instrukcji wewnętrznej określoną liczbę razy. <Zmiennej sterującej> nadawana jest najpierw wartość parametru <początek>, a po każdym wykonaniu instrukcji wewnętrznej <zmienna sterująca> zwiększana jest o wartość <kroku>. Iteracja jest wykonywana do momentu, gdy <zmienna sterująca> przekroczy wartość <granicy>. Gdy nie jest podany parametr <krok>, oznacza to, że jest on równy 1.

Instrukcja WHILE o postaci:

```
WHILE <warunek> DO <instrukcja>
```

służy do wykonania iteracji ze sprawdzaniem warunku na początku pętli.
Wykonanie instrukcji wewnętrznej jest powtarzane, dopóki <warunek> jest spełniony.

W instrukcji REPEAT, o postaci:

```
REPEAT <lista instrukcji> UNTIL <warunek>
```

sprawdzenie warunku następuje na końcu, a instrukcje wewnętrzne wykonywane są do chwili, gdy <warunek> zostanie spełniony.

W drugiej wersji instrukcji o postaci:

```
REPEAT <lista instrukcji> FOREVER
```

przerwanie powtarzania musi być dokonane przez jedną z instrukcji wewnętrznych.

Do instrukcji strukturalnych należy również podprogram o postaci:

```
SUBROUTINE <nazwa podprogramu>
                <lista instrukcji>
EXIT
```

Wykonanie tego podprogramu, który w odróżnieniu od procedury nie stanowi odrębnego segmentu programu następuje w wyniku użycia instrukcji EXECUTE.

OPERACJE NA DANYCH TEKSTOWYCH

Istnieje możliwość odwoływania się zarówno do całych zmiennych tekstowych, jak i do ich dowolnych fragmentów.

Odwołanie do podciągu (ang. substring) ma następującą postać:

T (p, d) lub TT (e, p, d)

gdzie:

T — zmienna pojedyncza typu TEXT, TT — tablica typu TEXT, p — numer początkowego znaku podciągu, d — długość podciągu, e — numer elementu tablicy.

W wyrażeniach logicznych może wystąpić relacja porównująca dane tekstowe. O wyniku porównania decyduje kolejność alfabetyczna (ściślej — wartości kodów binarnych poszczególnych znaków).

Operacje standardowej konwersji danych z tekstu lub na tekst mogą wystąpić w instrukcji podstawienia, instrukcjach redagujących oraz w instrukcjach wejścia-wyjścia. W instrukcjach redagujących i instrukcjach wejścia-wyjścia mogą wystąpić również konwersje sterowane formatami zawierającymi wzorce redakcyjne. Za pomocą formatów można również dokonywać kontroli postaci danych tekstowych, jak również dokonywać przekształceń tekstów polegających na dodawaniu lub opuszczaniu znaków.

Istnieją dwie instrukcje redagujące: EDIT i CODE. Instrukcja EDIT redaguje tekst wynikowy poprzez konkatenację stałych i zmiennych tekstowych, podciągów i danych innych typów, dokonując ich standardowej lub sterowanej konwersji. Istnieje możliwość zmiany kolejności redagowania tekstu poprzez odpowiednią tabulację. Przykładowo instrukcja:

```
EDIT NTXT : ='SUMA =', (X), FRM (CEN, IL)
```

przypisuje zmiennej NTXT wartość tekstową, składającą się z explicite podanego tekstu "SUMA = " i standardowo skonwertowanej zmiennej X, oraz — wartości zmiennych CEN i IL, zredagowanych według formatu o nazwie FRM.

Instrukcja CODE dokonuje rozpakowania informacji tekstowej i na jej podstawie nadania wartości wyspecyfikowanym zmiennym. Podczas rozpakowania tekstu dokonywane są odpowiednie konwersje (standardowe i sterowane formatem). Można dokonywać kontroli poprawności danych poprzez porównywanie ich z formatem. Możliwe są operacje opuszczania części tekstu i tabulacji. Przykładowo — instrukcja:

```
CODE (A,B,C) 86, FN (X,Y) := BUF (7,40)
```

dokonuje rozpakowania podciągu zmiennej BUF, pobierając z niego kolejno wartości zmiennych A,B,C (konwersja standardowa) i następnie — pomijając sześć znaków i pobierając wartości zmiennych X i Y (konwersja sterowana formatem wraz z kontrolą poprawności danych).

Instrukcje wejścia-wyjścia PUT i GET wykonują analogiczne operacje co instrukcje EDIT i CODE, z tym że zmienna tekstowa zastąpiona jest tu przez systemowy bufor wejścia-wyjścia. Dodatkowo specyfikowany jest strumień, za którego pośrednictwem dokonywane są przesłania pomiędzy buforem a urządzeniem zewnętrznym. Instrukcje wejścia-wyjścia zawierają operatory inicjujące kolejne transmisje, sterujące jednocześnie wysuwem linii i stron przez urządzenia drukujące.

W instrukcjach wejścia-wyjścia i w instrukcjach redagujących możliwe jest stosowanie konstrukcji analogicznej do pętli FOR, pozwalającej wykonywać funkcje tych instrukcji w sposób iteracyjny. W poniższym przykładzie za pośrednictwem strumienia LO wypisywana jest 100-elementowa tablica, po 10 elementów w linii:

```
I := 1;
PUT LO FOR K:=10,100,10
        (FOR I:=I,K ((TAB (I))) /)
```

ZBIORY

Język SIMBOL przewiduje różnorodne struktury zbiorów dyskowych, wygodne dla różnorodnych zastosowań. Istnieją następujące typy zbiorów różniących się strukturą i rodzajem dostępu:

Zbiór łańcuchowy (LINK) — zawierający rekordy powiązane w „łańcuch" za pomocą wskaźników adresowych. Możliwe jest dopisywanie, usuwanie lub modyfikowanie rekordów w dowolnym miejscu zbioru. Dostęp do rekordu jest sekwencyjny.

Zbiór indeksowany (IND) — zawiera rekordy, w których jedną lub więcej zmiennych tekstowych określono jako klucze. Możliwy jest dostęp do konkretnego rekordu poprzez podanie jego klucza (kluczy) lub dostęp według porządku określonego rosnącą kolejnością wybranego klucza.

Zbiór o dostępie bezpośrednim (RAN) — umożliwia dostęp do rekordu (odczyt lub zapis) poprzez podanie numeru rekordu w zbiorze.

Zbiór hierarchiczny — należy do jednego z trzech powyższych typów; z każdym jego rekordem związany może być łańcuch rekordów podporządkowanych (SLAVE) o budowie analogicznej do zbioru łańcuchowego. Rekordy podporządkowane mogą mieć z kolei własne rekordy podporządkowane tworząc tym samym struktury hierarchiczne (do trzech poziomów rekordów podporządkowanych). Dostęp do łańcucha rekordów podporządkowanych jest możliwy po odczytaniu (lub zapisaniu) rekordu nadrzędnego. Rekordy zbioru na jednym poziomie mają tę samą długość i strukturę. Rekordy występujące na różnych poziomach hierarchii mogą się różnić długością i strukturą.

Deklaracja zbioru zawiera określenie nazwy i typu zbioru oraz deklarację pól rekordów (głównych i podporządkowanych). Deklaracja zbioru indeksowanego zawiera ponadto specyfikację kluczy. Wartości zmiennych będących polami rekordu są odczytywane ze zbioru i zapisywane w zbiorze. Ponadto na polach rekordu można wykonywać wszelkie operacje, takie jak na zmiennych nie wchodzących w skład rekordu.

Instrukcje obsługi zbiorów są następujące:

OPEN — instrukcja otwarcia. Odszukuje zbiór na dysku, rezerwuje w pamięci obszar kontrolny dla obsługi danego zbioru. Kontroluje zgodność deklaracji zbioru z jego strukturą zapisaną na dysku.

CLOSE — instrukcja zamknięcia. Zwalnia obszar kontrolny w pamięci związany z zamykanym zbiorem.

EXTEND — dopisuje do zbioru nowy rekord. W zbiorze LINK i w łańcuchach SLAVE — dopisuje rekord na końcu łańcucha. W zbiorze IND poza zapisem rekordu wstawia jego klucze w tablice indeksów (zachowując rosnącą kolejność kluczy w tablicy), w zbiorze RAN — zapisuje rekord w miejsce określone numerem rekordu.

READ — instrukcja odczytu rekordu. W zbiorze LINK i w łańcuchach, SLAVE — odczytuje kolejny rekord w łańcuchu. W zbiorze IND odczytuje rekord zawierający określony klucz (klucze) lub kolejny rekord według określonej w tablicy indeksów sekwencji kluczy. W zbiorze RAN — odczytuje rekord o podanym numerze. W przypadku braku rekordu instrukcja READ przekazuje sterowanie pod etykietę określoną przez programistę.

REWRITE — zapisuje w zbiorze rekord ostatnio odczytany lub zapisany. Stosowana jest do modyfikacji danych rekordu — bez zmiany jego położenia na nośniku.

DELETE — usuwa ze zbioru ostatnio odczytany lub zapisany rekord.

INSERT — wstawia rekord w łańcuch bezpośrednio przed rekordem ostatnio odczytanym lub zapisanym.

REWIND — instrukcja „odwinięcia". Ustawia wskaźniki odczytu na pozycji pierwszego rekordu w łańcuchu lub na początek tablic indeksów.

Do obsługi zbiorów w SIMBOLU nie są konieczne żadne dodatkowe procesy odzyskiwania nieużytków (ang. garbage collection).

OBSŁUGA BŁĘDÓW

Błędy czasu wykonania programu w SIMBOLU mogą być obsługiwane systemowo lub programowo. Sposób obsługi błędów określony jest **instrukcją ONERROR**, która wskazuje miejsce w programie zawierające obsługę błędów lub przekazuje obsługę błędów do systemowego podprogramu.

IMPLEMENTACJA

Język SIMBOL został zaimplementowany na minikomputerze MERA 400 działającym pod nadzorem systemu operacyjnego SOM-3. Pakiet oprogramowania dostarczony użytkownikowi składa się z trzech elementów: kompilatora, biblioteki makrodefinicji i binarnej biblioteki podprogramów wykonawczych (biblioteka Run-Time). Kompilator generuje program w języku makroasemblera, co umożliwia umieszczanie w programach wstawek asemblerowych. Kompilator charakteryzuje się wygodną, czytelną diagnostyką błędów. Zawiera on szereg opcji, umożliwiających między innymi niezależną kompilację odrębnych segmentów programu oraz generowanie śledzenia kolejności wykonywania instrukcji (z wydrukiem wartości podstawianych w instrukcjach podstawiania). Proces tworzenia programu wynikowego obejmuje następujące etapy:

- generowanie przez kompilator kodu w języku makroasemblera
- generowanie przez makroasembler kodu binarnego przy wykorzystaniu biblioteki makrodefinicji
- ostateczna edycja przez konsolidator (ang. Link-Editor) z dołączeniem podprogramów biblioteki Run-Time (i ewentualnie innych).

Możliwe jest również wykorzystywanie istniejącej fortranowskiej biblioteki podprogramów oraz własnych bibliotek podprogramów napisanych w SIMBOLU, FORTRANIE lub makroasemblerze.

OPROGRAMOWANIE TOWARZYSZĄCE

Z językiem SIMBOL związane jest następujące oprogramowanie towarzyszące:

- program FMP (File Management Processor), umożliwiający między innymi alokację i wstępną inicjację symbolowych zbiorów dyskowych (niezbędny do przetwarzania zbiorów)
- monitor wielodostępu MAKS umożliwiający realizację w języku SIMBOL wielodostępnych systemów konwersacyjnych poprzez wykonywanie w jednym programie procesów współbieżnych, z których każdy związany jest z jednym terminalem; MAKS zawiera zestaw procedur umożliwiających synchronizację procesów współbieżnych
- STEP-400 — uniwersalny system zarządzania bazą danych oparty na ograniczonym modelu sieciowym CODASYL. Baza danych jest przetwarzana za pomocą programów w SIMBOLU, uzupełnionych komendami Języka Manipulacji Danymi. Baza danych jest niezależna od simbolowego systemu zbiorów.

Komputerowe systemy pomiarowe rozwinęły się na dobre wraz z rozpowszechnieniem minikomputerów, a szczególnie — mikroprocesorów. Jest to efekt dwóch zbieżnych tendencji: wyposażenia wszystkich przyrządów pomiarowych w układy cyfrowe o dużym stopniu scalenia oraz zwiększania się złożoności współczesnych metod pomiarowych.

Już obecnie rozwój tej techniki osiągnął taki poziom, że użytkownicy i producenci stanęli przed koniecznością standaryzacji różnych aspektów cyfrowego przetwarzania wyników pomiarów. Jednym ze skutków tego procesu było powstanie międzynarodowej normy magistrali łączącej urządzenia komputerowego systemu pomiarowego, znanej pod nazwą IEC-625 lub IEEE-488 i opatrzonej skrótem GPIB (ang. General Purpose Interface Bus). W poniższym artykule omówiono ogólne aspekty komputeryzacji zestawów pomiarowych, przedstawiając niektóre rozwiązania praktyczne. (Red.)

ELŻBIETA JEZIERSKA-ZIEMKIEWICZ
JAN RYŻKO

Instytut Maszyn Matematycznych
Warszawa

Programowane systemy pomiarowe

Programowane systemy pomiarowe są obecnie budowane przeważnie na podstawie tzw. IEC-bus (znany także jako GPIB, tj. General Purpose Interface Bus) sprzęg opracowany przez firmę HEWLETT-PACKARD (pod nazwą HP-IB) wprowadzony do zaleceń IEC [1]) jako sprzęg IEC-625-1, a uprzednio przyjęty jako standard IEEE 488-1975 (z rewizją w 1978 r.). American National Standards Institute przyjął normę IEEE w roku 1976 [3].

Z danych sekretariatu IEC wynika, że w roku 1982 ok. 300 firm produkowało ok. 3000 przyrządów wyposażonych w ten sprzęg. Były to zarówno przyrządy pomiarowe (multimetry, kalibratory, sondy itp.), jak również sterowniki systemów pomiarowych oparte na mikro- i minikomputerach. Wprowadzenie tanich i łatwych w eksploatacji sterowników systemów pomiarowych zapoczątkowało gwałtowny rozwój automatyzacji miernictwa w laboratoriach badawczych i w produkcji, przy testowaniu i kontroli jakości wyrobów.

[1]) IEC — International Electrotechnical Commission

CHARAKTERYSTYKA MAGISTRALI IEEE-488

Standard IEEE-488 umożliwia połączenie 15 urządzeń (w tym — co najmniej jednego tzw. sterownika systemu) w programowany system pomiarowy. Magistrala systemu składa się z 16 linii:
— ośmiu dwukierunkowych linii danych
— trzech linii czasowego sterowania transmisją
— pięciu linii sterujących.

Zdefiniowane są następujące funkcje urządzeń:
SH (Source Handshake) — sterowanie nadawaniem
AH (Acceptor Handshake) — sterowanie odbiorem

T lub **TE** (Talker, Extended Talker) — nadajnik
L lub **LE** (Listener, Extended Listener) — odbiornik
SR (Service Request) — żądanie obsługi
RL (Remote, Local) — praca zdalna lub lokalna
PP (Parallel Poll) — zapytywanie
DC (Device Clear) — zerowanie urządzenia
DT (Device Trigger) — wyzwalanie urządzenia
C (Controller) — kontroler [2]).

Urządzenia mogą być wyposażone w dowolny podzbiór tych funkcji [8].

Ograniczeniami magistrali IEEE-488 są:
— maksymalna całkowita długość kabla równa 20 m
— maksymalna szybkość transmisji przez linię równą 1 M bit/s
— brak gwarancji na zgodność funkcjonalną urządzeń ze względu na dowolność treści komunikatów i format informacji.

Obecnie czyni się próby znormalizowania kodów i formatu informacji 1.

Każde urządzenie (przyrząd pomiarowy, sterownik itp.) musi być przystosowane do realizacji co najmniej jednej z trzech funkcji: nadawcy, odbiorcy lub kontrolera (T, L lub C). Nadawca może przekazywać dane przez magistralę do innych modułów systemu, a odbiorca odbiera dane nadawane przez inny moduł systemu.

Urządzenie może być również dwukierunkowe — nadawczo-odbiorcze. Urządzenie zarządzające szyną (np. sterownik) realizuje wszystkie funkcje, tj. T, L i C. W celu realizacji algorytmu pomiarowego przesyła sekwencje programujące do urządzeń dołączonych do systemu, nadzoruje wykonanie programu, zbiera i interpretuje wyniki pomiarów. Oprogramowanie takiego sterownika jest albo stałe (dla zamkniętych systemów pomiarowych), albo generowane w języku wysokiego poziomu, najczęściej w BASICU.

RODZAJE PROGRAMOWANYCH SYSTEMÓW POMIAROWYCH

Komputeryzacja aparatury pomiarowej (spektrometrów, mikroskopów elektronowych itp.) umożliwia zbieranie wielu informacji pomiarowych w krótkim czasie, ich przetwarzanie i zobrazowywanie (np. w postaci histografów), zapewniając większą dokładność i szybkość pomiarów. Mikroprocesor zarządza aparaturą pomiarową (przez specjalizowane układy — na ogół cyfrowo-analogowe) oraz umożliwia dołączenie przyrządu do większego systemu pomiarowego, przez łącze standardowe transmisji szeregowej lub magistralę typu IEEE-488. Można wymienić wiele przykładów takiej aparatury [2, 5, 6, 7].
Do komunikacji z operatorem służą przeważnie uproszczone klawiatury funkcyjne i wyświetlacze. Mikroprocesor pracuje pod kontrolą stałego programu, bez możliwości tworzenia dodatkowego oprogramowania przez użytkownika. Monitor ekranowy ma często większe możliwości graficzne i wyspecjalizowaną klawiaturę. Wyjścia cyfrowe umożliwiają dołączenie przyrządu do złożonego systemu pomiarowego.

Przykłady komputeryzacji przyrządów pomiarowych

Niezwykle popularnymi urządzeniami przy projektowaniu, wytwarzaniu i obsłudze sprzętu elektronicznego są oscyloskopy. Muszą one zapewnić możliwość oglądania, mierzenia i przetwarzania szerokiej gamy przebiegów elektrycznych.

Wielofunkcyjnym przyrządem wyposażonym w sprzęg IEEE-488 jest oscyloskop typu 1980A/B firmy HEWLETT-PACKARD [6]. Sterowanie cyfrowe umożliwia łatwą obsługę, automatyczny dobór zakresu, pełne zaprogramowanie pomiarów, pamiętanie przebiegów w postaci cyfrowej oraz sprzętowy i programowy rozwój systemu.

Przy zdalnym programowaniu przez wejście IEC można zadawać wszystkie mierzone parametry, wyświetlać napisy na ekranie itp. Stany przyrządu, wyniki pomiarów i cyfrowo zapamiętane przebiegi mogą być przesyłane do komputera sterującego w celu dalszego przetwarzania. Sprzęg IEEE-488 umożliwia połączenie oscyloskopu z innymi przyrządami w zautomatyzowany system badawczy. Opracowane podprogramy mogą być przechowywane i używane wielokrotnie. Dodatkowe wyposażenie, jak np. pamięć przebiegów cyfrowych typu 198GOA umożliwia uchwycenie, wyświetlenie i przesłanie do komputera powtarzalnych przebiegów o częstotliwości do 100 MHz i jednorazowych o częstotliwości do 4 MHz, bez interwencji operatora. W ten sposób można przechowywać w postaci cyfrowej szereg przebiegów standardowych w celu automatycznego badania i porównywania.

Ze względu na wzrost wymagań w zakresie dokładności pomiarów stosuje się częstą i dokładną kalibrację przyrządów i systemów kontroli automatycznej. Większość kalibratorów umożliwia uzyskanie dokładności 1 na 100 000 i wymaga wyspecjalizowanej obsługi.

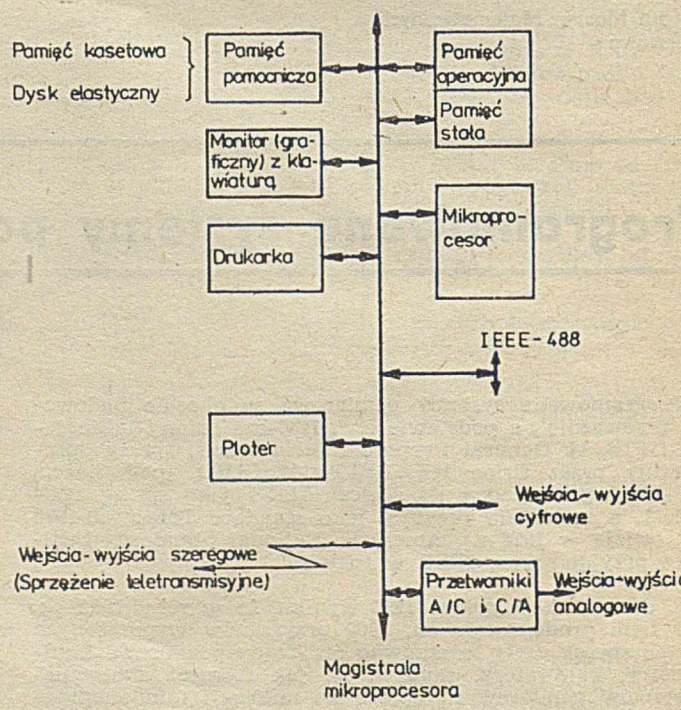
Dzięki użyciu mikroprocesorów w kalibratorze typu 5440A firmy JOHN FLUKE możliwe jest uproszczenie działania i zwiększenie dokładności [2]. Przyrząd przeprowadza kalibrację z dokładnością czterech części na milion i może być obsługiwany przez pracowników bez specjalnego przeszkolenia. Ponadto, jest mniej wrażliwy na zmiany otoczenia i może być używany np. na liniach produkcyjnych lub odległych stacjach obsługi. Wykorzystane są tu trzy mikroprocesory: do sterowania części analogowej, do zarządzania współpracą z użytkownikiem i do kierowania działaniem całego systemu i sprzężeniem z systemami zewnętrznymi.

Mikroprocesorowe sterowniki systemów pomiarowych

Szerokie zastosowanie standardu IEEE-488 stwarza możliwość zestawiania systemów pomiarowych z przyrządów, sterowników (mini- i mikrokomputerowych) oraz urządzeń peryferyjnych. Inteligentne sterowniki umożliwiają redukcję, analizę, pamiętanie i przetwarzanie wyników pomiarów, a także zarządzanie systemem pomiarowym.

Na rysunku 1 przedstawiono uproszczony schemat programowanego systemu pomiarowego. W zastosowaniach, w których wymagana jest duża częstotliwość pomiarów lub złożone przetwarzanie uzyskanych wyników, zastępuje się mikrokomputer systemem minikomputerowym lub łączy go hierarchicznie z komputerem o większej mocy obliczeniowej.

Firma HEWLETT-PACKARD instaluje specjalizowane systemy pomiarowe budowane w oparciu o urządzenia pomiarowe i sterowniki (komputery) własnej produkcji. Do



Rys. 1. Uproszczony schemat programowanego systemu pomiarowego

[2]) Autorzy proponują stosowanie terminu k o n t r o l e r jako nazwę funkcji, a s t e r o w n i k — jako nazwę urządzenia

sterowania używane są stołowe komputery (ang. desktop computer) serii HP 9800 z wbudowanymi urządzeniami peryferyjnymi, pamięciami operacyjnymi i stałymi. Przykładowo — model 9825 B zawiera:

- 23 K bajty pamięci RAM
- 32-znakowy wyświetlacz na diodach świecących
- pamięć taśmową (kasety o pojemności 256 K bajtów)
- klawiaturę alfanumeryczną
- klawiaturę funkcyjną
- sprzęg szeregowy V24
- sprzęg IEEE-488

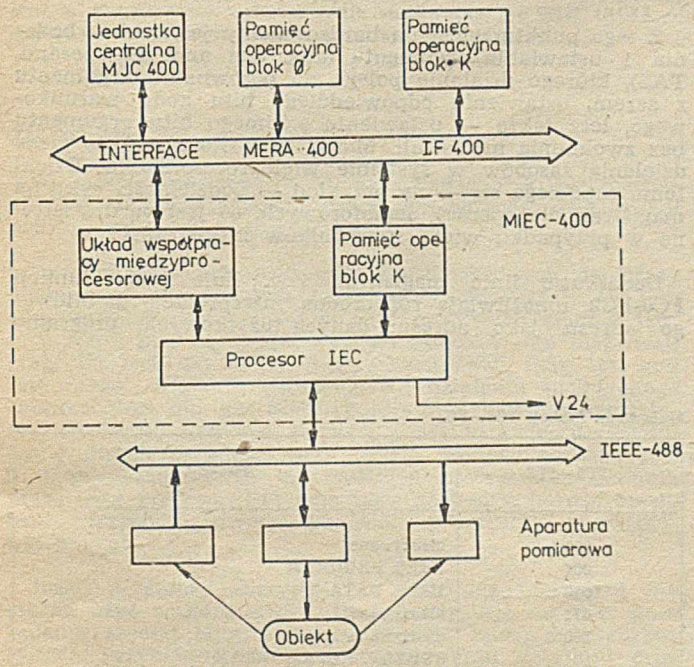
oraz ma możliwość sterowania 16 wejściami-wyjściami binarnymi.

Najbardziej rozbudowany model jest wyposażony w monitor graficzny wraz z oprogramowaniem, drukarkę termiczną 80-znakową i umożliwia dołączenie plotera i dysku. Podstawowym językiem programowania jest BASIC z rozszerzeniami dotyczącymi grafiki sterowania urządzeń pomiarowych oraz — z instrukcjami pętli (REPEAT, WHILE, LOOP), warunkowymi (IF, SELECT, CASE) itp.

Ze względu na pewne niedoskonałości normy IEEE-488 na poziomie kodów i formatów, zwiększające koszty zastosowania tych systemów, firma TEKTRONIX wprowadziła normę gwarantującą zgodność programową wszystkich własnych przyrządów oraz opracowała modułową serię przyrządów uniwersalnych, które mogą być programowane i sterowane przez specjalizowany sterownik (komputer) TM 5000 [1].

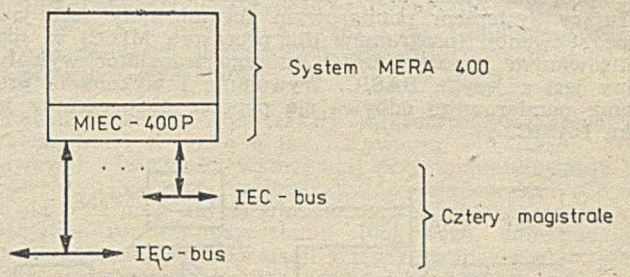
Norma IEEE-488 określa stronę mechaniczną (złącze, kabel), elektryczną (poziomy sygnałów oraz sposób ich wysyłania i odbioru) i funkcjonalną (zadania realizowane przez sprzęg przyrządu). Brak jest natomiast normy dotyczącej zasad porozumienia się, co grozi komplikacjami przy rozwoju ilościowym i jakościowym przyrządów. Dlatego w normie firmy TEKTRONIX określono część wspólną komunikatów, elementy komunikacji i ich kombinacje oraz protokoły. Upraszcza to i przyspiesza pisanie oprogramowania użytkowego, programy są krótsze i lepiej udokumentowane.

Konsekwencją stosowania tej normy jest wprowadzenie przez firmę TEKTRONIX nowej serii programowanych przyrządów TM 5000, dzięki czemu znacznie zmniejszono nakłady na opracowanie oprogramowania. Wszystkie te urządzenia współpracują z magistralą IEEE-488, są sterowane mikroprocesorami i zawierają złożone procedury testowania i diagnostyki. Mikroprocesory służą do przetwarzania wyników pomiarów, zapamiętywania przebiegu badań i śledzenia stanu napięć i prądów zasilających.



Rys. 2. Schemat blokowy systemu MERA 400 do celów pomiarowych

Jednocześnie z serią TM 5000 opracowano sterownik typu 4041, dostosowany do systemów kontrolno-pomiarowych, oparty na trzech mikroprocesorach. Oprogramowanie, tworzone przez wyspecjalizowany zespół konstruktorów zawiera m.in. rozszerzony BASIC oraz umożliwia wieloprogramowanie (współbieżność operacji wejścia-wyjścia i obliczeń).



Rys. 3. Zwarty system pomiarowy

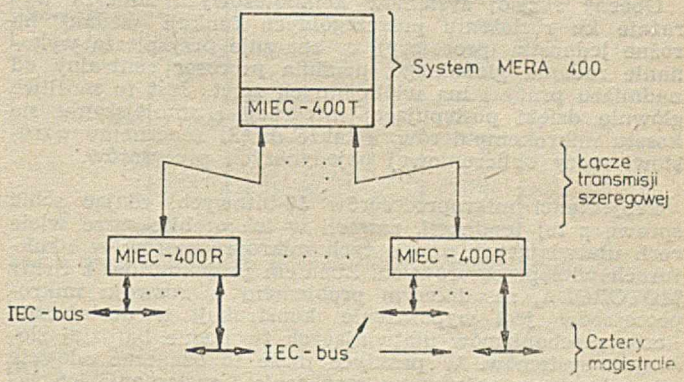
Systemy te są szeroko stosowane zarówno w laboratoriach jak i liniach produkcyjnych. W pierwszym przypadku jedna osoba jest jednocześnie programistą i operatorem, a w drugim — automatyzacja eliminuje całkowicie człowieka. Zestawy rozbudowane zawierają monitor ekranowy, drukarkę lub ploter oraz grafoskop i umożliwiają rozszerzenie oprogramowania.

* * *

Z przedstawionego przeglądu wynika, że korzystne jest prowadzenie prac naukowo-badawczych w następujących kierunkach:

— mikrokomputeryzacja aparatury pomiarowej
— modularne, programowane systemy pomiarowe oparte na strukturach mikroprocesorowych
— wyposażanie systemów minikomputerowych w moduły umożliwiające ich wykorzystanie w systemach pomiarowych.

Przedsiębiorstwa produkujące urządzenia pomiarowe (np. UNIMA, MERATRONIK, EUREKA) wprowadzają mikroprocesory do opracowywanych przez siebie przyrządów pomiarowych. Z drugiej strony tworzy się moduły minikomputerowe (MERA 60 i MERA 400) do sprzężenia z magistralą IEEE-488.



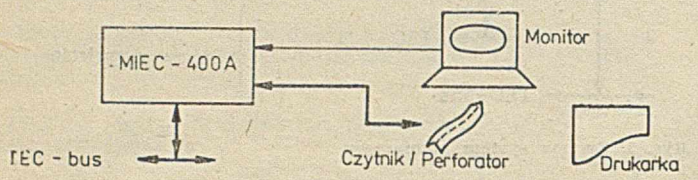
Rys. 4. Hierarchiczny system pomiarowy

W Instytucie Maszyn Matematycznych opracowano moduł MIEC-400 do systemu minikomputerowego MERA 400, zrealizowany w technice mikroprocesorowej [4, 9]. MERA 400 wyposażona w moduł MIEC-400 może sterować czterema niezależnymi magistralami IEEE-488. Schemat blokowy takiego zestawu przedstawiono na rysunku 2. Funkcjonalnie moduł MIEC-400 jest procesorem peryferyjnym ze specjalizowanym do celów pomiarowych zbiorem instrukcji (emulowanych przez mikroprocesor) i realizuje następujące funkcje sprzęgu IEEE-488:

SH1 — sterowanie nadawaniem
AH1 — sterowanie odbiorem
T2 — nadawca

L2 — odbiorca
SR1 — żądanie obsługi
C1—C5 — kontroler systemu.

Oprogramowanie systemu pomiarowego działa pod kontrolą systemów operacyjnych CROOK-3 i CROONIX; zawiera język BASIC rozszerzony o konstrukcje umożliwiające programowanie algorytmów pomiarowych. Algorytm sterujący systemem pomiarowym formułowany jest jako zbiór sekwencji (programów dla procesora MIEC) w specjalizowanym języku LASEQ, którego translator wywoływany jest z języka BASIC. Wywołanie i wykonanie programu pomiarowego odbywa się poprzez podprogramy języka BASIC.



Rys. 5. Prosty system pomiarowy

Moduł MIEC-400 umożliwia tworzenie urozmaiconych konfiguracji systemów pomiarowych (rys. 3—5).

LITERATURA

[1] Balland J. C.: Vers une nouvelle generation d'instruments GP-IB — la serie TM 5000. Electronique Industrielle, No. 24, 1981
[2] Eccleston L. and al.: Calibrator Brings Record Accuracy Even to Production and Repair. Electronics, 8 September 1982
[3] IEEE Standard Digital Interface for Programmable Instrumentation. IEEE Std 488/1978
[4] Moduł Interface IEC typ MIEC-400. Informacja techniczna, Instytut Maszyn Matematycznych, Warszawa, 1982
[5] Rhodes-Burke R.: Retrofitting for Signature Analysis Simplified. Hewlett-Packard Journal, January 1982
[6] Risley W. B.: Oscilloscope Measurement System Is Programmable and Autoranging. Hewlett-Packard Journal, September 1982
[7] Roberts R. J.: An Integrated Test Set for Microwave Radio Link Baseband Analysis. Hewlett-Packard Journal, April 1982
[8] Schwartz Ph.: Pour optimiser vos mesures, utiliser le bus IEEE-488. Electronique Industrielle, No. 12, 1981
[9] System MERA 400. Moduł Interface IEC typ MIEC-400. Dokumentacja Techniczno-Ruchowa, t. IV, cz. 31, Instytut Maszyn Matematycznych, Warszawa, 1982.

JANUSZ ZALEWSKI
Warszawa

Użycie mikroprocesorów 16-bitowych w przetwarzaniu rozłożonym

Obecny rozwój systemów komputerowych zmierza wyraźnie ku rozłożeniu poszczególnych funkcji (zadań) na różne jednostki (procesory), co znacznie przyspiesza wykonanie zadania globalnego, uwalnia procesor centralny od nadmiaru pracy i ma wiele innych zalet. Jest to możliwe głównie dzięki postępującej integracji i zmniejszaniu się kosztu mikrokomputerów, a także dzięki znacznemu wzrostowi mocy obliczeniowej pojedynczych procesorów.

Projektanci mikroprocesorów 16-bitowych, zdając sobie sprawę z tej tendencji, zawarli w ich architekturze wiele cech ułatwiających użycie tych mikroprocesorów w strukturach wieloprocesorowych. Zdaniem projektantów z firmy MOTOROLA, zasadniczym problemem w rozwoju mikroprocesorów jest wyposażenie konstrukcji w dostateczną liczbę mechanizmów ułatwiających tworzenie bardziej złożonych systemów. W praktyce może to oznaczać odejście od systemów z pojedynczą magistralą i zastąpienie ich zestawami wielomagistralowymi. Innym zagadnieniem o kluczowym znaczeniu jest określenie zasad współpracy procesorów w systemie rozłożonym, a następnie implementacja tych zasad w architekturze.

Poniżej omówimy wybrane właściwości mikroprocesorów MC 68000, Z8000, iAPX 186 i NS 16032, sprzyjające tworzeniu zestawów wieloprocesorowych.

MOTOROLA 68000

Mikroprocesory 8-bitowe mogą bezpośrednio komunikować się tylko z urządzeniami synchronicznymi, tzn. zdolnymi do wymiany danych w określonym odcinku czasu. Komunikacja z urządzeniami asynchronicznymi wymaga dodatkowych układów sprzęgających. Mikroprocesor MOTOROLA 68000 ma natomiast linie sterujące do bezpośredniej komunikacji z urządzeniami asynchronicznymi, co jest znacznym ułatwieniem w budowie zestawów wieloprocesorowych.

Z tego punktu widzenia bardzo ważny jest rozkaz badania i ustawiania argumentu (ang. test and set operand, TAS), którego działanie polega na porównaniu argumentu z zerem, ustawieniu odpowiedniego bitu kodu warunkowego, lecz także — ustawieniu siódmego bitu argumentu bez zwolnienia magistrali. Można to wykorzystać do przydzielania zasobów w systemie wieloprocesorowym. Dzięki temu zapobiega się — na przykład — podzielności cykli na magistrali dla operacji semaforowych, co jest bardzo istotne w przypadku wielu sterowników magistrali.

Dodatkowe linie magistrali, tzw. linie kodu funkcji FC0-FC2, umożliwiają różnorodną interpretację 23-bitowego adresu jako adresu: danych użytkowych, programu

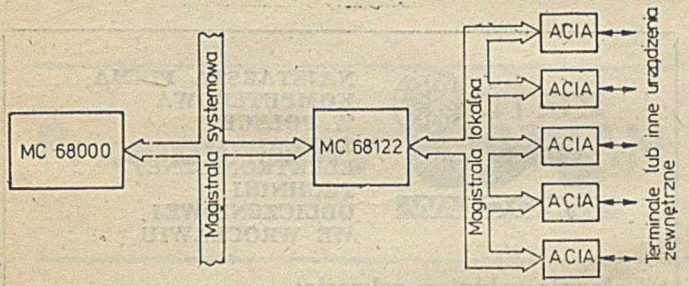
Linie kodu funkcji FC0—FC2

FC0—FC2	Funkcja
000	nieużywane
001	USER PROGRAM
010	USER DATA
011	nieużywane
100	nieużywane
101	SUPERVISOR PROGRAM
110	SUPERVISOR DATA
111	INTERRUPT ACKNOWLEDGE

użytkowego, danych systemowych lub programu systemowego (tab.). W praktyce oznacza to rozszerzenie obszaru adresowego mikroprocesora do czterech 16-bitowych segmentów. Dodatkowy bit BGACK (ang. bus grant acknowledge) umożliwia adresowanie i użytkowanie magistrali przez sterowniki podległe (ang. slaves), a nie tylko nadrzędne (ang. masters).

Niemniej, omówione właściwości dotyczą systemów rozłożonych funkcjonalnie, lecz w zasadzie skupionych przestrzennie (ang. tightly coupled). Przykładem takiego zestawu jest połączenie magistrali dwóch mikroprocesorów MC 68000 przez układy bezpośredniego dostępu do pamięci MC 68450.

Specjaliści firmy MOTOROLA uważają jednak, że znacznie większe korzyści osiągnie się stosując zestawy faktycznie rozłożone geograficznie, w tzw. konfiguracjach słabo lub umiarkowanie powiązanych (ang. loosely, moderately coupled). Z tego względu wprowadzono szereg mikroukładów o specjalnym przeznaczeniu, pełniących funkcje różnego rodzaju sterowników. Jednym z najważniejszych jest inteligentny sterownik urządzeń zewnętrznych MC 68120 (ang. intelligent peripheral controller). W jego skład wchodzi 8-bitowa jednostka centralna, 2 KB pamięci ROM, 128 KB pamięci RAM, 16-bitowy czasomierz, sprzęg szeregowy i 23 linie wejściowo-wyjściowe. Tak złożony sterownik może z powodzeniem przejąć ciężar wymiany informacji wejściowo-wyjściowej, uwalniając główny procesor od zajmowania się komunikacją. Odpowiednio zaprogramowany układ MC 68120, nazwany sterownikiem skupienia terminali (ang. cluster terminal controller, MC 68122) służy jako urządzenie sprzęgające grupę terminali (dołączonych do magistrali lokalnej) z magistralą systemową (rys. 1). Dzięki użyciu tego sterownika liczba przerwań sterownika macierzystego zmniejsza się ponad 60 razy w jednostce czasu, co powoduje znaczne zmniejszenie narzutu systemowego na procesor.



Rys. 1. Przykładowe wykorzystanie sterownika MC 68122 (ACIA — Asynchronous Communication Interface Adapter)

ZILOG 8000

W mikroprocesorze Z8000 wprowadzono kilka mechanizmów umożliwiających pracę wieloprocesorową wokół jednej magistrali, zwanej Z-bus.

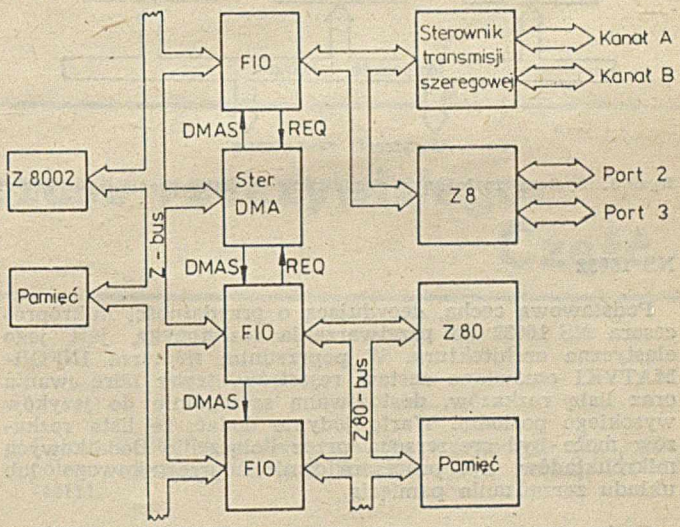
Do podstawowych mechanizmów niezbędnych do pracy wieloprocesorowej należą operacje synchronizacji. W tym celu wykorzystuje się jeden z 16 możliwych stanów tzw. linii stanu procesora, ST0-ST3, do zablokowania magistrali (ang. bus lock) po wykonaniu rozkazu TSET (ang. test and set). Jego działanie (zbliżone do działania omówionego rozkazu TAS dla MC 68000) polega na zbadaniu zawartości określonej komórki pamięci i jej ustawieniu. Wykonanie dwóch czynności przy użyciu jednego rozkazu jest o tyle pożądane, że uniemożliwia przerwanie działania przez inny procesor. Sytuacja taka następuje bardzo często przy korzystaniu dwóch lub wielu procesorów ze wspólnej pamięci.

Jeżeli procesory korzystają ze wspólnego zasobu i nie można użyć rozkazu TSET, to wykorzystuje się tzw. linie żądania zasobu (ang. resource request lines) oraz rozkaz MREQ. Przykładowo — przy użytkowaniu wspólnej drukarki przez trzy procesory, po bezskutecznej próbie przejęcia rozkazem MREQ sterowania zajętą drukarką — próbę ponawia się.

Istotną cechą rodziny Z8000 jest także możliwość znacznego zrównoleglenia zadań (ang. cooperative transactions) na poziomie elementów i wykonywania części tego samego programu przez różne jednostki (procesory). Przykładowo — wykorzystując możliwości adresowe procesora centralnego, procesor dodatkowy (ang. extended processing unit, EPU) może przejąć wykonywanie niektórych rozkazów, działając równolegle z procesorem centralnym (CPU). Dopiero w chwili pobierania przez CPU kolejnego rozkazu dla nie gotowego jeszcze EPU, generuje on sygnał STOP do procesora centralnego.

Pewne szczegóły architektury mikroprocesora Z8000 zaprojektowano z myślą o przekazywaniu wiadomości, podstawowej właściwości wymaganej przy komunikowaniu się procesów. Ponieważ przekazywanie wiadomości odbywa się na ogół przy użyciu bloków danych, mikroprocesor ma specjalne rozkazy przesyłania blokowego, umożliwiające transmisję bloków informacji między pamięcią a urządzeniami wejścia-wyjścia, bez użycia tradycyjnych pętli programowych. Przykładowo — rozkaz powtarzanego wejścia INDR (ang. input, decrement and repeat) wykorzystujący trzy rejestry, polega na określeniu adresu urządzenia wejściowego (rejestr 1), zaadresowaniu pamięci (rejestr 2), przesłaniu do niej słowa 16-bitowego, zmniejszeniu adresu o 2 oraz zmniejszeniu zawartości licznika słów o 1 (rejestr 3).

Bardzo ważną właściwością mikroprocesora są tzw. transmisje przelotowe (ang. flyby transfers), polegające na superszybkim bezpośrednim dostępie do pamięci, lecz bez przejmowania danych przez sterownik DMA, co w przybliżeniu dwukrotnie przyspiesza transmisje DMA. Jest to możliwe dzięki użyciu specjalnego mikroukładu, tzw. FIO (FIFO Input-Output), który jest buforem typu FIFO i stanowi podstawowy układ do sprzęgania procesorów (rys. 2).



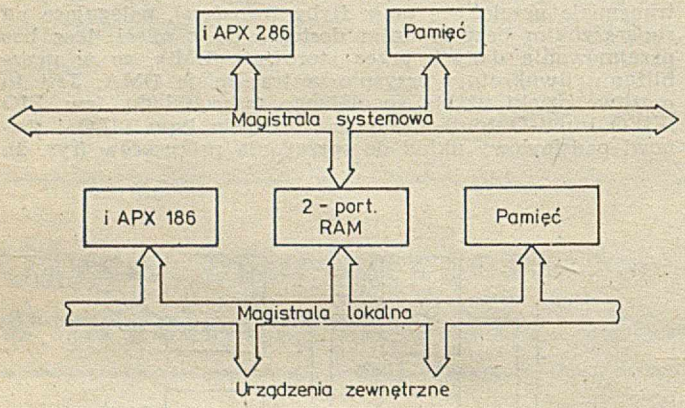
Rys. 2. Rola układu FIO w systemie wieloprocesorowym

iAPX 186

Podobną filozofię przyjęła firma INTEL opracowując nowy 16-bitowy mikroprocesor iAPX 186, stanowiący zintegrowaną odmianę mikroprocesora typu 8086, omówionego w numerze 4/83 (warto zaznaczyć, że obecnie procesor ten nazywa się iAPX 86, a systemy wieloprocesorowe z jego użyciem, oparte na magistrali MULTIBUS i sterownikach oraz arbitrach magistrali, typu 8288 i 8289, są już dość powszechnie stosowane). W mikroukładzie iAPX 186 zawarto — oprócz 16-bitowej jednostki centralnej — ok. 10 innych układów, stanowiących uprzednio oddzielne elementy. Są to m.in.: generator zegarowy (do 16 MHz) i trzy 16-bitowe czasomierze, sterownik DMA, układ przerwań, układ selekcji (ang. chip select/ready generation). W praktyce jest to więc pierwszy 16-bitowy mikroprocesor jednoukładowy. Każdy z wymienionych podukładów ma własny zestaw rejestrów sterujących, umieszczonych w przeznaczonym na to 256-bajtowym bloku pamięci.

Mikroprocesor iAPX 186 jest zatem tak rozbudowany, że dla pełnienia funkcji w systemach rozłożonych nie wymaga żadnych indywidualnych właściwości. Przykładowo — jego przystosowanie do roli węzła sieci mikrokomputerowej wynika z bardzo elastycznej architektury, obejmującej uprzednio wymienione elementy. Zawarty w układzie blok DMA może sterować pracą dwóch niezależnych kanałów między pamięcią a obszarem wejścia-wyjścia, lecz także — wewnątrz dowolnego z tych obszarów, co umożliwia np. pominięcie układów dekodujących przy transmisji wejściowo-wyjścowej. Wbudowany sterownik przerwań ma wszystkie funkcje układu przerwań typu 8259A, a zdefiniowany z góry wektor przerwań zwiększa 1,5-krotnie szybkość odpowiedzi w porównaniu z mikroprocesorem INTEL 8086. Różnorodne możliwości funkcjonalne czasomierzy i układu selekcji uzupełniają zestaw niezbędnych cech tego mikroprocesora.

W porównaniu ze swym poprzednikiem (INTEL 8086) iAXP 186 jest wzbogacony o 10 nowych rozkazów, m.in. przesyłania blokowego, tzw. natychmiastowego mnożenia, przesyłania zawartości wszystkich rejestrów na stos oraz pobierania ich w całości ze stosu. Skojarzone z procesorem macierzystym iAPX 286, mikroprocesory iAPX 186, mogą tworzyć bardzo skuteczny, rozłożony system przetwarzania (rys. 3).



Rys. 3. System rozłożony, zbudowany z mikroprocesorów iAPX

NS 16032

Podstawową cechą, decydującą o przydatności mikroprocesora NS 16032 do przetwarzania rozłożonego, jest jego elastyczna architektura. W poprzednim numerze INFORMATYKI omówiono zestaw rejestrów, tryby adresowania oraz listę rozkazów, dostosowaną szczególnie do języków wysokiego poziomu. Warto jedynie dodać, że lista rozkazów może być rozszerzona przez dołączenie dodatkowych mikroukładów, np. procesora zmiennoprzecinkowego lub układu zarządzania pamięcią.

Cechą wyróżniającą mikroprocesor NS 16032 wśród pozostałych procesorów tej klasy, pod względem dostosowania do przetwarzania rozłożonego, jest jego system operacyjny. Przewidziano w nim m.in. rzecz najważniejszą, mianowicie, przezroczystość konfiguracji sprzętu dla oprogramowania. Znaczy to, że mechanizm wzajemnej komunikacji zadań (procesów) nie zależy od tego, czy zadania rezydują w tym samym procesorze, czy w różnych procesorach (węzłach), ani też — od sposobu połączenia tych procesorów. Zresztą, sam system operacyjny może być podzielony na wiele zadań, które łatwo rozłożyć na sieć procesorów, komunikujących się przy użyciu tych samych mechanizmów.

System operacyjny mikroprocesora NS 16032 ma budowę modułową i składa się z jądra oraz zadań zarządzania zasobami (ang. resource manager tasks). Jądro steruje wykonywaniem zadań i dostępem do zasobów na poziomie maszyny. Zadanie zarządzania zasobami (zarządca zasobu) realizuje dostęp na wyższym, dostępnym dla programisty poziomie abstrakcji. Przykładowo — zarządca plików decyduje o dostępie zadań do dysku, a zarządca obszaru pamięci realizuje mechanizm pamięci wirtualnej dla pozostałych zadań.

Komunikacja między zadaniami jest realizowana przez jądro przy użycu tzw. obwodów (ang. circuits). Każdy obwód jest połączeniem logicznym, umożliwiającym przekazywanie wiadomości do innych zadań lub do jądra i może być rozciągnięty na całą sieć. Celowi temu służy jedno z zadań zwane zarządcą sieci. Jeżeli porozumiewające się zadania połączone obwodem rezydują w różnych węzłach, to w transmisji pośredniczy zarządca sieci w każdym węźle.

Oprócz zarządzania zadaniami, system operacyjny NS 16032 zawiera jednolity mechanizm zarządzania obiektami (ang. object management mechanism). Według terminologii, którą posługuje się firma NATIONAL SEMICONDUCTOR, obiekt jest egzemplarzem pewnego abstraktu zdefiniowanego przez użytkownika (an instance of some user defined abstraction), jak np. plik, skorowidz czy urządzenie. Istotną cechą tego mechanizmu jest możliwość dostępu do obiektów bez względu na to, gdzie one się znajdują i skąd pochodzi żądanie dostępu. Operacje określające zachowanie obiektów są realizowane przez zadania zwane zarządcami obiektów, odpowiadające określonym ich typom. Przykładowo — tzw. zarządca skorowidzów (ang. directory manager) zapewnia obsługę nie hierarchicznej struktury plików. W systemie rozłożonym, gdzie w zasadzie nie ma możliwości utworzenia globalnego skorowidza, np. w postaci drzewa, taki program jest niezbędny.

* * *

Powyżej opisano jedynie wybrane problemy związane z użyciem mikroprocesorów 16-bitowych w systemach rozłożonych. Ze względu na bardzo zróżnicowane architektury poszczególnych mikroprocesorów, trudno w tym opisie zastosować jednolite podejście. Spośród mikroprocesorów nie omawianych w tym artykule, ważną rolę w sieciach komputerowych odgrywa LSI-11. Choć ma on może najmniej cech immanentnie związanych z przetwarzaniem rozłożonym, jest obecnie używany w tym celu znacznie częściej niż inne procesory.

ZETO WROCŁAW

NAJSTARSZA FIRMA KOMPUTEROWA W POLSCE ZAKŁAD ELEKTRONICZNEJ TECHNIKI OBLICZENIOWEJ WE WROCŁAWIU

oferuje usługi w zakresie:

- **doradztwo w obszarze zastosowań informatyki (rozwój i wdrażanie, eksploatacja, konserwacja sprzętu)**
- **projektowanie, programowanie oraz wdrażanie systemów informatycznych**
- **doradztwo i projektowanie systemów cyfrowych**
- **serwis techniczny minikomputerów i urządzeń klimatyzacyjnych**
- **projektowanie ośrodków obliczeniowych oraz szkolenie użytkowników komputerów i minikomputerów**
- **udostępnianie mocy obliczeniowych komputerów JS RIAD oraz ODRA 1305 wsadowo i konwersacyjnie wraz z przygotowaniem maszynowych nośników informacji**
- **wykonywanie systemów operacyjnych i oprogramowania narzędziowego minikomputerów**

UWAGA: Od 1984 r. ZETO WROCŁAW przystępuje do świadczeń jako GENERALNY DOSTAWCA minikomputerów SM-4

Zakład Elektronicznej Techniki Obliczeniowej

ul. Ofiar Oświęcimskich 7/13, 50-069 WROCŁAW
Tel. centr. 44-54-31 do 37. Teleks: 0712533 ZETO PL
Zgłoszenia przyjmuje: DZIAŁ MARKETINGU

EO/815/K/83

Projekt normy FORTRANU może się wydawać czymś nie tak interesującym, by należało publikować go w INFORMATYCE. Trzeba więc na wstępie wyjaśnić, że chodzi tu jedynie o uzupełnienia norm międzynarodowych języka FORTRAN (ISO 1539-1980, PN-76/T-42111) w zakresie programowania systemów czasu rzeczywistego.

Właściwie od początku stosowania FORTRANU w systemach przemysłowych wiele problemów mających identyczną budowę programowano najróżniejszymi sposobami, używając całkowicie odmiennych struktur tego języka, a często nawet różnych języków niskiego poziomu. Początkowo dotyczyło to tylko analogowych i cyfrowych operacji wejścia-wyjścia oraz operacji bitowych, a w mniejszym stopniu — informowania o czasie i zarządzania plikami. Ostatnio pojawiła się natomiast nowa grupa zagadnień wymagających unormowania — zarządzanie zadaniami.

Spośród różnych możliwych podejść do ujednolicenia FORTRANU w zastosowaniach przemysłowych, najbardziej racjonalna wydaje się próba uzgodnienia standardowych procedur do wykonywania określonych funkcji, gdyż nie prowadzi to do jakichkolwiek zmian w istniejących kompilatorach. Tą drogą poszły prace organizacji Instruments Society of America, która jako pierwsza, już w 1972 roku, opublikowała normy dotyczące podstawowych procedur FORTRANU przemysłowego. Obecna propozycja obejmuje i uaktualnia wcześniejsze normy, lecz idzie znacznie dalej w kierunku standaryzacji współpracy wielu zadań.

Wydaje się, że w chwili obecnej, kiedy produktywność programistów staje się wąskim gardłem w wielu zastosowaniach, a FORTRAN w zasadzie nie ma groźnych konkurentów w systemach przemysłowych (może poza ADĄ) — standaryzacja odpowiada potrzebom użytkowników i prezentowana norma może odegrać istotną rolę w tej dziedzinie. (Red.)

Przemysłowy FORTRAN czasu rzeczywistego. Część 1

Artykuł zawiera omówienie projektu normy opracowanej przez Komitet Techniczny nr 1 Międzynarodowej Grupy Roboczej Purdue ds. Przemysłowych Systemów Komputerowych (EWICS). Część pierwsza obejmuje definicję procedur dla zastosowania FORTRANU do sterowania procesów przemysłowych. Oryginalny dokument (raport IPW/EWICS TC 1, 1/82 — Industrial Real-Time FORTRAN — IRTF) powstał przy udziale przedstawicieli Austrii, Holandii, Norwegii, Polski [1], RFN i W. Brytanii.

GENEZA NORMALIZACJI FORTRANU

Język FORTRAN został opracowany przez firmę IBM w 1955 r. Od tego czasu stał się najszerzej używanym językiem wysokiego poziomu dla zastosowań naukowych, a obecnie dostępne są w tym języku obszerne i efektywne biblioteki programów użytkowych.

FORTRAN został przedstawiony do normalizacji w roku 1967 i ostatecznie znormalizowany przez Międzynarodową Organizację Normalizacyjną (ISO) w 1972. Aktualna definicja FORTRANU jest zawarta w raporcie [9], którego polski odpowiednik opublikowano jako normę PN-76/T-42111.

Wymagania specjalne

Szerokie wykorzystanie i wypróbowana przydatność FORTRANU do zastosowań naukowych wkrótce doprowadziły do jego użycia w przemysłowych systemach czasu rzeczywistego. Takie zastosowania wymagają operacji specjalnych (operacji czasu rzeczywistego), operowania ciągami bitów i — środków dla wejścia-wyjścia procesu. Wymienione operacje można zrealizować jedną z następujących dróg:

- FORTRAN pozostaje podstawowym językiem do obliczeń arytmetycznych oraz do wprowadzania i wyprowadzania danych za pośrednictwem standardowych urządzeń zewnętrznych; operacje specjalne realizuje się przez elementy spoza składni FORTRANU
- pełny język czasu rzeczywistego pozostaje w ramach składni FORTRANU, łącznie z operacjami specjalnymi; oznacza to, że operacje te muszą być fortranowskimi procedurami funkcyjnymi lub niefunkcyjnymi.

Pierwsze podejście prowadzi do typowego języka czasu rzeczywistego mającego proste, ale mocne, narzędzia programowe do operacji specjalnych. Przy takim rozszerzaniu

[1] Przedstawicielem Polski w Komitecie Technicznym nr 1 EWICS jest mgr inż. Kazimierz Maliszewski (Przemysłowy Instytut Automatyki i Pomiarów), który przetłumaczył wspomniany dokument; pełna lista autorów obejmuje 18 nazwisk.

języka projektanci starają się wykorzystać wszystkie możliwości użytego systemu operacyjnego czasu rzeczywistego, wskutek czego rozszerzenia stają się zależne od konkretnego systemu. Tak było szczególnie we wczesnych opracowaniach tego rodzaju [5, 7, 8]. Niedogodności tej uniknięto dopiero w języku PROCOL, stworzonym we Francji dla komputerów produkcji krajowej. Ma on bardzo rozwinięte środki do programowania w czasie rzeczywistym [3].

Wszystkie języki z rozszerzeniami wykraczającymi poza składnię FORTRANU wymagają specjalnych kompilatorów do translacji.

Wybranie drugiego podejścia i pozostanie w ramach składni FORTRANU, daje tę korzyść, że pierwszą kompilację i sprawdzenie programu użytkowego można wykonać na każdym komputerze, dla którego istnieje kompilator tego języka. Z drugiej strony — podejście używające procedur funkcyjnych i niefunkcyjnych prowadzi do nieco kłopotliwej manipulacji dodatkowymi wywołaniami podprogramów typu SUBROUTINE i FUNCTION i związanymi z nimi parametrami. Można to jednak uznać za niewielką niedogodność, ponieważ wielu użytkowników FORTRANU jest przyzwyczajonych do tego rodzaju programowania.

FORTRAN przemysłowy warto porównać z innymi językami czasu rzeczywistego, jak:
— przemysłowy BASIC czasu rzeczywistego
— języki zaprojektowane specjalnie do zastosowań w czasie rzeczywistym, jak: PEARL, RTL/2, itp.
— ADA [15].

Różne odmiany BASICA czasu rzeczywistego są bardzo łatwe do opanowania i użycia, nadają się bardzo dobrze dla oprogramowania prostych, małych problemów. Można je zrealizować stosunkowo łatwo zarówno na dużych, jak i małych systemach komputerowych.

Języki o specjalnym charakterze przeznaczone dla systemów czasu rzeczywistego dostarczają bardzo mocnych środków programowania, dlatego też nadają się szczególnie do problemów dużych i złożonych. Ich realizacja (kompilator i system operacyjny czasu rzeczywistego) jest stosunkowo kosztowna.

W 1980 r. pojawiła się jeszcze jedna możliwość, tzn. język ADA. Jest on bardziej uniwersalny niż specjalne języki czasu rzeczywistego, a jego rozmiary i złożoność sytuują go w tej samej grupie kosztów realizacji. ADA ma bardzo dobre właściwości synchronizacji (rendezvous), natomiast nieco ograniczone możliwości szeregowania zadań.

Języki podobne do FORTRANU przemysłowego zrealizowane pierwszą metodą są często zbliżone do specjalnych języków czasu rzeczywistego, natomiast języki zrealizowane drugą metodą są pod wieloma względami usytuowane pomiędzy BASICEM i specjalnymi językami czasu rzeczywistego. FORTRAN jest więc alternatywnym rozwiązaniem w stosunku do dwóch wymienionych grup językowych. W rezultacie, język zgodny z drugim podejściem musi być stosunkowo prosty, co oznacza, że ze względu na łatwość uczenia się i programowania liczba, i złożoność dodatkowych operacji powinna być ograniczona.

Rozwój prac nad normą

W celu zapobieżenia powstaniu wielu niezgodnych języków czasu rzeczywistego, w 1970 r. na uniwersytecie Purdue (USA) utworzono Grupę Roboczą dla Normalizacji Przemysłowych Języków Komputerowych pod przewodnictwem Teodora J. Williamsa. Po połączeniu z inną organizacją (1973), grupa ta przyjęła nazwę „Międzynarodowa Grupa Robocza w Purdue ds. Przemysłowych Systemów Komputerowych" (IPWICS — International Purdue Workshop on Industrial Computer Systems). Komitet FORTRANU tej organizacji był od początku bardzo aktywny. Z różnych powodów do opracowania języka wybrano drugie z wymienionych podejść, ze wszystkimi specjalnymi operacjami utrzymanymi w ramach składni języka FORTRAN. Zdając sobie sprawę, że niemożliwe jest opracowanie jednej złożonej normy w rozsądnym czasie, wspomniany komitet zdecydował się rozwiązać problem w postaci serii norm. Program pracy został przedstawiony i przyjęty w organizacji normalizacyjnej ISA (Instrument Society of America), gdzie utworzono Komitet Techniczny SP61. Komitet ten w stosunkowo krótkim czasie opracował pierwszą propozycję FORTRANU czasu rzeczywistego, która została zatwierdzona i opublikowana jako norma ISA S61.1 (1972) [10]. Praca zawiera następujące grupy operacji specjalnych:
— trzy wywołania (instrukcja CALL) oraz fortranowską instrukcję STOP dla zarządzania stanem uruchamianych współbieżnie „programów" [2]
— sześć wywołań dla wejścia-wyjścia procesu
— pięć funkcji całkowitych do operowania ciągami bitów (stosowanego do liczb całkowitych, traktowanych jako ciągi bitów).

W rok później opublikowano drugą normę uzupełniającą — ISA S61.2 (1973), która zawiera następujące grupy operacji specjalnych:
— siedem wywołań do operowania niezredagowanymi plikami o bezpośrednim dostępie
— funkcję logiczną do badania pojedynczego bitu liczby całkowitej
— dwa wywołania dla ustawiania i zerowania pojedynczego bitu liczby całkowitej.

Wymienione normy ISA dzięki użyciu w różnych krajach zdobyły duże uznanie (z wyjątkiem propozycji dla operacji na plikach). Wczesne opracowanie i opublikowanie tych norm przez amerykański Komitet FORTRANU organizacji IPWICS i ISA było ważnym krokiem w programowaniu procesów przemysłowych.

Projekty norm ISA dalej rozwijano i ostatecznie zatwierdzono jako normy ANS/ISA. Tak więc, ANS/ISA S61.1 [1] zawiera wszystkie części poprzednich norm z wyjątkiem zarządzania plikami, które jest tematem normy ANS/ISA S61.2 (1978) [2]. Istotną różnicą normy S61.1 z 1976 r. w stosunku do wcześniejszych opracowań jest to, że nie dopuszcza ona wywołań podprogramów bez oczekiwania na zakończenie procedury. Zmiana ta została dokonana dla zachowania zgodności z operacją wywołania zewnętrznej procedury macierzystego języka FORTRAN [9].

Ponieważ opracowanie ISA S61.1 zawiera zbiór wywołań procedur, które zyskały szerokie uznanie, można uważać ten dokument za podstawowy Przemysłowy FORTRAN Czasu Rzeczywistego organizacji IPWICS. Zawiera on tylko trzy wywołania i instrukcję STOP do zarządzania równoległymi zadaniami (programami). Ponieważ potrzebne są bardziej złożone mechanizmy, amerykański Komitet FORTRANU IPWICS opracował projekt uzupełniającej normy dotyczącej zarządzania zadaniami — ISA S61.3 [12]. Równolegle do tej pracy, grupa robocza „Prozess-FORTRAN" niemieckich stowarzyszeń VDI/VDE [3] opracowała pełną propozycję pn. „Prozess — FORTRAN 75", wydaną jako projekt normy VDI/VDE [17].

„Prozess FORTRAN 75" zawiera 11 wywołań dla zarządzania zadaniami równoległymi i stanowi ograniczone, ale mocne narzędzie programowania operacji czasu rzeczywistego. Przetwarzanie ciągów bitów i pojedynczych bitów jest podobne jak w normie ISA S61.1 (1976) — różni się jedynie opisem. Wprowadzono dodatkowe funkcje całkowite do przesunięcia arytmetycznego i cyklicznego oraz wywołanie podprogramu do zmiany bitu. Wejście-wyjście do procesu jest praktycznie takie samo, jak w ISA S61.1 (1976), jedynie normalizacja analogowych wejść-wyjść prowadzi do ograniczenia normy. Zarządzanie plikami jest zrealizowane inaczej niż w normie ISA S61.2 (1977), która odpowiadała ówczesnym propozycjom.

W 1976 r. członkowie grupy roboczej „Prozess-FORTRAN" VDI/VDE i inni specjaliści z Europy utworzyli Komitet Techniczny nr 1 ds. Przemysłowego FORTRANU Czasu Rzeczywistego organizacji EWICS [4]. Komitet postanowił opracować pełną propozycję przemysłowego FORTRANU czasu rzeczywistego.

---

[1]) W normie tej stosuje się termin „zadanie" zamiast terminu „program" używanego w [1, 2, 10, 11].

[2]) VDI/VDE — Verein Deutscher Ingenieure/Verband Deutscher Elektrotechniker

[3]) EWICS — European Workshop on Industrial Computer Systems, ówczesna nazwa — Purdue Europe

Do końca 1977 r. współpraca pomiędzy Komitetem amerykańskim i europejskim była dość luźna, odbywało się tylko jedno wspólne zebranie w roku — spotkanie plenarne organizacji IPWICS. Tak więc oba komitety pracowały do pewnego stopnia niezależnie, wskutek czego ich opracowania były pod kilkoma względami całkowicie różne. Dlatego kierownictwo organizacji IPWICS i EWICS, ISO/TC97/SC5/WG1, a także i ECMA TC8 [5]) zażądało opracowania propozycji wspólnych, a najlepiej — całkowicie identycznych. Do osiągnięcia tego celu należało w znacznym stopniu zintensyfikować współpracę.

Poczynając od połowy 1978 r. w większości spotkań brał udział przynajmniej jeden członek drugiej zainteresowanej organizacji. Dzięki intensywnej współpracy międzynarodowej osiągnięto istotne postępy, co zaowocowało opracowaniem niniejszej normy. Równocześnie w celu zakończenia swego programu prac, komitet ISA SP61 opracował projekt normy S61.3 [12], w dużym stopniu zgodny z omawianym opracowaniem.

Aby ułatwić osiągnięcie wspólnego celu w postaci powstania jednej międzynarodowej normy FORTRANU do sterowania procesami przemysłowymi, w ISA podjęto kilka działań:

- zalecono ANSI [6]), aby wycofać normę ANS/ISA S61.1 [1] jako przedmiot prac dla ISO/TC97/SC5/WG1 [7]); akcja ta została podjęta
- zalecono ANSI, aby nie proponować normy ANS/ISA S61.2 [2] jako przedmiotu prac WG1
- zdecydowano, aby projektu normy S61.3 [12], jeżeli zostanie zatwierdzony, nie przesyłać do ANSI w celu wydania normy krajowej.

Poniższa norma będzie wymagała w przyszłości dostosowania do nowych warunków, aby umożliwić rozwój języka, a także rozwój systemów operacyjnych czasu rzeczywistego. Z czasem będą zatem wprowadzane odpowiednie uzupełnienia.

MECHANIZMY WIELOPROGRAMOWEJ PRACY W CZASIE RZECZYWISTYM

W pierwszym rozdziale projektu normy opisano grupę wywołań procedur dostępnych w programie użytkowym, odnoszących się do pracy wieloprogramowej, a zwłaszcza — do działania w czasie rzeczywistym. Dla wszystkich wywołań z tego rozdziału ich działanie uważa się za niepodzielne, tj. powinny zachowywać się tak, jakby nie były przerywane.

INFORMACJA O DACIE I CZASIE

Dla programowania w warunkach czasu rzeczywistego użytkownik musi mieć dostęp do zmiennych przechowujących czas w systemie nadzorczym. Zmienne te są odczytywane przez wywołania systemowe. Jednoznaczne określenie czasu wymaga wskazania chwili doby, łącznie z kompletną datą i przyjętym kalendarzem, określającym punkt zerowy czasu. Dzięki wywołaniu podprogramu DATIM otrzymuje się tę pełną informację (dla kalendarza gregoriańskiego).

Odczyt daty i czasu

Postać wywołania jest następująca:
CALL DATIM (t1)
gdzie:
— t1 oznacza tablicę całkowitą [8]), której pierwszych osiem elementów zawiera czas bezwzględny, podany przez zegar czasu rzeczywistego systemu nadzorczego w chwili wykonania wywołania; kolejne elementy mają znaczenie następujące: jednostki zegara podstawowego, milisekundy (0—999), sekundy (0—59), minuty (0—59), godziny (0—23), dzień (1—31), miesiąc (1—12), rok.

Odczyt jednostek zegara

Wykonanie tego wywołania umożliwia programowi użytkowemu uzyskanie bieżącej wartości czasu zegara systemowego, wyrażonej w podstawowych jednostkach zegara. Postać wywołania jest następująca:

CALL CLOCK (j,k1,k2)

gdzie:

j — określa zmienną całkowitą lub element tablicy całkowitej, w którym bieżąca wartość czasu będzie umieszczona jako dodatnia liczba całkowita (j jest zliczane do maksymalnej wartości określonej przez **k2**, zerowane i ponownie zliczane)

k1 — jest liczbą podstawowych jednostek zegara na sekundę i pobiera wartość całkowitą dostarczoną przez system (argument ten powinien być zmienną całkowitą lub elementem tablicy całkowitej)

k2 — określa maksymalną liczbę, jaką może być **j**; jest to wartość całkowita dostarczana przez system (argument ten powinien być zmienną całkowitą lub elementem tablicy całkowitej).

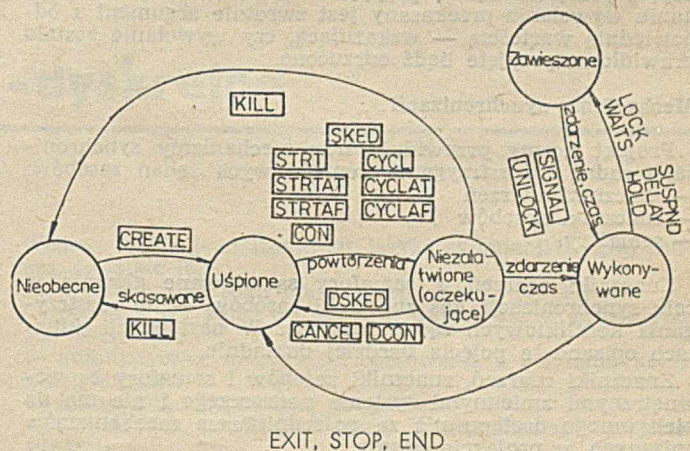
Wielkość okresu zliczania [9]) argumentu **j** jest określona argumentem **k2** (okres zliczania = k2+1). W rzeczywistości będzie on zwykle równy okresowi zliczania sprzętowego licznika realizującego zegar.

OGÓLNE ZASADY PRACY ZADAŃ

Stany i przejścia

W dowolnej chwili zadanie jest w jednym i tylko jednym stanie. Działania wykonane przez system nadzorczy, przez inne zadania lub przez samo rozpatrywane zadanie — mogą spowodować jego przejście z jednego stanu do drugiego. Takie przejścia są wykonywane natychmiast, tzn. zakłada się, że nie zajmują wcale czasu.

Ten matematyczny model zadania można przedstawić na schemacie stanów (rys.), w którym stany są węzłami przedstawionymi jako koła, zaś przejścia są wykreślone przez strzałki skierowane od jednego węzła do drugiego [14].



Model stanów i schemat przejść

System wielozadaniowy zawiera pewną liczbę równoległych zadań i można go opisać przez zbiór rozdzielnych, ale podobnych schematów. Używając trójwymiarowego obrazu, podobne schematy nakłada się jeden na drugi — tak, aby identyczne stany poszczególnych schematów pokrywały się.

Przejścia międzystanowe są — ogólnie rzecz biorąc — spowodowane przez wywołanie podprogramów, wystąpienie zdarzeń lub przekroczenie limitów czasowych. Nazwa, postać i interpretacja wywołań podprogramów są znormalizowane i opisane w niniejszym dokumencie.

---

[5]) ECMA — European Computer Manufacturers Association

[6]) ANSI — American National Standards Institute

[7]) Working Group 1 — grupa robocza w podkomitecie nr 5 komitetu technicznego nr 97 ISO

[8]) W niniejszym tekście używa się równoważnie określeń c a ł k o w i t y i t y p u c a ł k o w i t e g o — w ślad za tekstem angielskim i dla symetrii do pozostałych typów

[9]) W oryginale: modulus

Zadanie opisano przez model matematyczny zilustrowany schematem stanów na rysunku. Jest on zgodny z następującymi zasadami:
1. Przejścia są niedwuznaczne, tj. dla danego bodźca w danym stanie zadanie może przejść tylko do jednego możliwego nowego stanu.
2. Przejścia są wykonywane natychmiastowo, tj. w czasie równym zero.
3. Zadanie istnieje w danej chwili tylko w jednym stanie.
4. Model stanów opisuje zachowanie zadań z punktu widzenia programisty użytkowego.

Na schemacie stanów zastosowano następujące oznaczenia dla przejść:
- duże litery w ramce oznaczają wpływ na zadanie wywierany przez inne zadanie (wywołanie podprogramu w jednym zadaniu ma taki wpływ na wskazane zadanie)
- duże litery bez ramki oznaczają wpływ zadania w stanie WYKONYWANE na siebie
- małe litery oznaczają warunki wykonania wskazanych przejść przez system nadzorczy.

Zgodnie z czwartą zasadą nie starano się opisać działań systemu nadzorczego, niewidocznych dla programisty użytkowego. W związku z tym stan WYKONYWANE odnosi się do procesora wirtualnego zadania. Dla stanu zadania jest nieistotne, czy procesor fizyczny został mu przydzielony, czy też wykonanie system nadzorczy chwilowo wstrzymał, wskutek ograniczonej dostępności procesorów fizycznych i niskiego priorytetu zadania. W ten sposób model jest dostosowany zarówno do komputerów wielojak i jednoprocesorowych.

Wielokrotne wywołania aktywacji

Różne zadania mogą wydawać sprzeczne wywołania przejść dla zadania wskazanego. Może się to zdarzyć w czasie normalnego działania, jak i w warunkach błędu, ponieważ stan zadania będącego przedmiotem wywołania jest nieznany w chwili, gdy inne zadanie wykonuje wywołanie.

Zgodnie z wymaganiem jednoznaczności stanu w jednej chwili, rozróżnia się przejścia międzystanowe i wywołania żądające takich przejść. Wywołania przejść są przyjmowane przez system nadzorczy, który stosuje własną strategię szeregowania przy obsłudze takich żądań [10].

Zależnie od dostępnych zasobów, jak np. miejsce w wewnętrznych tablicach systemu nadzorczego, wywołanie żądające przejścia będzie przyjęte lub odrzucone. Po wykonaniu wywołania przekazany jest zwrotnie argument z odpowiednią wartością — wskazującą, czy wywołanie zostało prawidłowo przyjęte bądź odrzucone.

Mechanizmy synchronizacji

Projekt normy przewiduje trzy mechanizmy synchronizacji zadań i rozstrzygania konfliktowych żądań zasobów:
— znaczniki zdarzeń
— znaczniki zasobów
— semafory.

Znaczniki zdarzeń i semafory są używane głównie w celu synchronizacji, zaś znaczniki zasobów — do rozstrzygania konfliktowych żądań zasobów. W następnych punktach opisano te pojęcia bardziej dokładnie.

Znaczniki zdarzeń, znaczniki zasobów i semafory są wewnętrznymi zmiennymi systemu nadzorczego i nie ma do nich innego dostępu niż za pośrednictwem mechanizmów opisanych w projekcie normy.

Znaczniki zdarzeń

Przy zarządzaniu zadaniami współbieżnymi konieczne jest wiązanie określonych zadań z określonymi zdarzeniami. Zdarzenia te mogą być zdarzeniami zewnętrznymi albo wewnętrznymi. Zdarzenia zewnętrzne mają określoną naturę fizyczną (np. zwarcie styków), jednak związek pomiędzy zewnętrznym zdarzeniem i jego znacznikiem pozostaje poza zakresem normy. Zdarzenie wewnętrzne powstaje wskutek określonego działania programu (por. opis CALL POST). Wybór znacznika zdarzenia następuje przez podanie jego numeru z zakresu 1 do n, gdzie n jest zależne od procesora. Znaczniki zdarzeń mają dwa stany: WŁĄCZONY i WYŁĄCZONY. Dołączenie zdarzeń do zadań realizuje się przez wywołanie podprogramów CALL SKED, CALL CON, CALL SUSPND i CALL HOLD, opisane w części 2.

Znacznik zdarzenia jest ustawiany w stan WŁĄCZONY przez wystąpienie zdarzenia wewnętrznego lub zewnętrznego. Jeżeli z tym zdarzeniem jest związane jedno lub więcej zadań, to system nadzorczy spowoduje rozpoczęcie lub kontynuację wykonania każdego z dołączonych zadań. Znacznik zdarzenia jest przełączany w stan WYŁĄCZONY wskutek bezpośredniego działania programu lub przez system nadzorczy, przy wykonaniu obsługi zadań związanych ze znacznikiem zdarzenia. Stan znaczników zdarzeń może się zmieniać tylko przez wywołania procedur (zdefiniowane w normie) oraz przez system nadzorczy, który obsługuje zdarzenia.

Znaczniki zdarzeń mogą być również ustawione w stan WYŁĄCZONY przez określone działanie programu, CALL CLEAR, oraz mogą być maskowane i odmaskowane. Ponadto wartość znacznika zdarzenia może być badana przez funkcję logiczną TESTEM.

Znaczniki zasobów

Prosty sposób rozstrzygania konfliktowych żądań różnych zasobów został ustanowiony przez mechanizm znaczników zasobów, który umożliwia jednemu i tylko jednemu zadaniu używać zasobu w danej chwili. Znaczniki zasobów wybiera się przez wskazanie liczbowego identyfikatora z zakresu od 1 do n, gdzie n jest zależne od procesora.

Znaczniki zasobów mają dwa stany: ZAMKNIĘTY i OTWARTY. Projekt normy nie określa, co może być uważane za zasób. Związanie zasobu ze znacznikiem zasobów pozostawia się odpowiedzialności użytkownika. Jeżeli użytkownik chce zarezerwować jakiś zasób wyłącznie do wykonywanego zadania, to wybiera znacznik zasobu dla tego zasobu i wykonuje wywołanie CALL LOCK. Jeżeli znacznik zasobu jest w tej chwili w stanie OTWARTY, to zmieni swój stan na ZAMKNIĘTY, rezerwując odpowiadające mu urządzenie wyłącznie dla tego zadania. Rezerwacja ta jest później odwoływana przez zadanie, wskutek wykonania wywołania EXIT lub UNLOCK.

Gdyby zadanie próbowało, za pomocą procedury LOCK, zamknąć już zamknięty znacznik zasobu, to system nadzorczy przeniesie to zadanie do stanu ZAWIESZONE, w którym pozostanie ono dotąd, aż znacznik zasobu nie zostanie otwarty przez inne zadanie. Jeżeli kilka zadań oczekuje aż znacznik zasobu zostanie OTWARTY, to tylko jedno z nich będzie wybrane do wykonania przez system nadzorczy. Jest to główna różnica w stosunku do mechanizmu znaczników zdarzeń, gdzie wszystkie zadania oczekujące na znacznik zdarzenia są przenoszone do stanu WYKONYWANE, jeżeli znacznik zdarzenia jest ustawiany w stan WŁĄCZONY.

Znacznik zasobu może być także ZAMKNIĘTY przez wywołanie funkcji logicznej TLOCK.

Semafory

Semafory stanowią koncepcję synchronizacji użyteczną w przypadkach, gdzie pożądany jest bardziej rozwinięty mechanizm niż ten, który jest dostępny dzięki znacznikom zasobów i zdarzeń. Semafory są wybierane przez odwołanie się do liczbowego identyfikatora z zakresu od 1 do n, gdzie n jest zależne od procesora. W przeciwieństwie do znaczników zdarzeń i zasobów, semafor ma wartość całkowitą s. Wartość s semafora musi być nadana wstępnie i może być ustawiona później przez wywołanie CALL PRESEM.

Jeżeli zadanie ma czekać aż wartość semafora będzie większa lub równa określonej wartości, to używa ono wywołania CALL WAITS, które powoduje, że system nadzorczy sprawdza, czy dane j jest większe niż s. W takim przypadku zadanie wywołujące jest przenoszone do stanu ZAWIESZONE, w którym pozostaje do chwili, gdy wartość semafora zostanie zwiększona przez inne zadanie do wartości większej lub równej j. Jeżeli j nie jest większe od s, to wartość s jest zmniejszona o j i zadanie wywołujące kontynuuje swe wykonanie.

Wywołanie CALL SIGNAL powoduje, że system nadzorczy zwiększa o j wartość s określonego semafora. Zwiększ-

[10] W niniejszej pracy szeregowanie (ang. scheduling) należy rozumieć jako zaplanowanie; zgodnie z tym, co powiedziano w poprzednim punkcie, szeregowanie nie odnosi się do współzawodnictwa związanego z procesorem rzeczywistym

szenie to może przenieść zadanie oczekujące pod tym semaforem ze stanu ZAWIESZONE do stanu WYKONYWANE, jeżeli s staje się większe lub równe j zawieszonego zadania. Jeżeli kilka zadań oczekuje pod tym semaforem i kandyduje do wykonania po zwiększeniu s o j, to porządek, w którym zadania te przechodzą do stanu WYKONYWANE, zależy od procesora.

Przez nadanie s określonych wartości i wybór różnych wartości dla j, możliwe są różnorodne złożone mechanizmy synchronizacji i rozwiązania sporów o zasoby. Jako uzupełnienie opisanych wyżej wywołań, wartość semafora może być odczytana przy użyciu funkcji całkowitej IRDSEM.

LITERATURA

[1] ANS/ISA S61.1 (1976) Standard. Industrial Computer System FORTRAN Procedures for Executive Functions, Process Input-Output, and Bit Manipulation. Instrument Society of America, 1976

[2] ANS/ISA S61.2 (1978) Standard. Industrial Computer System FORTRAN Procedures for Fils Access and the Control of File Contention. Instrument Society of America, 1978

[3] Bonnard P.: PROCOL: A Programming System Adapted to Process Control. IFAC Congress, Paris, 1972

[4] Brinch Hansen P.: Operating System Principles. Prentice-Hall, 1973

[5] DDP-516 OLERT, Programmer's Referance Manual, Honeywell Inc., Doc. No. 1300 72069 A, EM-602, 1969

[6] Dijkstra E. W.: Cooperating Sequential Processes. In „Programming Languages" (V. Genuys, ed.), pp 43—112. American Press, New York, 1968

[7] Hohmeyer R. E.: CDC 1700 FORTRAN for Process Control. IEEE Transactions on IECI, Vol. 15, No. 2, December 1968

[8] INDAC 8. Digital Equipment Corporation, 1969

[9] International Standard ISO 1539 — Programming Languages: FORTRAN (ISO 1539 — 1980)

[10] ISA S61.1 (1972) Standard. Industrial Computer System FORTRAN Procedures for Executive Functions and Process Input-Output. Instrument Society of America, 1972

[11] ISA S61.2 (1973) Draft Standard: Industrial Computer System FORTRAN Procedures for Handling Random Unformatted Files, Bit Manipulation, and Date and Time Information. Instrument Society of America, 1973

[12] ISA S61.3 (Aug. 1978) Draft Standard: Industrial Computer System FORTRAN Procedures for the Management of Independent Interrelated Tasks. Instrument Society of America, 1978

[13] ISO Recommendation R 1539, Programming Language FORTRAN, ISO, Geneva (Switzerland), 1972

[14] Pettersen O.: Management of Parallel Activities in Real-Time FORTRAN. State Model and State Transitions. 2nd edition. Minutes of the Purdue Europe Spring Meeting 1977 at ISPRA CCR Euratom, Vol. II. IRIA, Rocquencourt, 1977

[15] Reference Manual for the Ada Programming Language. United States Department of Defense, July 1980. (Document ISO/ /TC97/SC5 N619)

[16] Shaw A. C.: The Logical Design of Operating Systems. Prentice-Hall, 1974

[17] VDI/VDE 3556, Prozess-FORTRAN 75, eine Erweiterung von FORTRAN für Prozessrechner-Anwendungen. VDI/VDE-Richtlinie (Entwurf). VDI/VDE — Gesellschaft für Mess- und Regelungstechnik, Düsseldorf, 1978.

Tłumaczył:
KAZIMIERZ MALISZEWSKI

TERESA WILCZEK

Centrum Obliczeniowe SGGW-AR
Warszawa

Aspekty użytkowe języków symulacyjnych (2)

Symulacja ciągła i ciągło-dyskretna

Języki programowania symulacji ciągłej przeznaczone są do modelowania systemów, których stan aktualny może być zdefiniowany jako zależność funkcyjna od stanów przeszłych (historii) oraz od oddziaływań otoczenia. Interpretacja zachowania się takich systemów w postaci matematycznych równań stanu narzuca jednoznaczny sposób opisu ich struktury fizycznej. Języki symulacji ciągłej pozwalają na zaprogramowanie modelu ciągłego w sposób nie różniący się zasadniczo od rozwiązania stosowanego na maszynie analogowej. Dlatego też różnice między językami tej grupy wynikają głównie z odmiennych możliwości funkcjonalnych i szczegółów programowych.

Języki wczesnego rozwoju technik symulacyjnych operują pojęciami przejętymi bezpośrednio z modelowania analogowego i wymagają budowy schematów blokowych badanych układów, przy czym każdy element schematu odpowiada określonemu operatorowi. Rozwój techniki cyfrowej oraz powstawanie nowych metod programowania wywarły duży wpływ na koncepcję języków symulacji ciągłej. Wykorzystując cechy języków proceduralnych, takich jak np. FORTRAN, budowano symulatory pozwalające na zaprogramowanie problemu bezpośrednio na podstawie równań modelu matematycznego, bez przedstawienia go w postaci bloków funkcjonalnych. Żaden jednak ze znanych języków symulacji ciągłej nie spełnia, w wystarczającym stopniu, wymagań Komitetu Oprogramowania Modelowania (The Simulation Software Committee). Według opinii tego Komitetu języki symulacji ciągłej muszą posiadać [6]:
— prostą postać zdań opisujących układ
— podstawowy zbiór operatorów, dający się w prosty sposób rozszerzać
— automatyczny mechanizm całkowania i ustalania kolejności obliczeń
— kompletną diagnostykę programu
— rozszerzone możliwości pracy z urządzeniami wejścia-wyjścia
— możliwość zmiany parametrów
— możliwość współpracy z jednym ze znanych języków algorytmicznych
— możliwość pracy konwersacyjnej
— możliwość budowy programów zorientowanych na opis modelu za pomocą bloków lub równań.
Przedstawiona w tabeli 1 kolejność języków do symulacji ciągłej odpowiada, w ogólnym zarysie, ich rosnącym możliwościom obliczeniowym.

Jednym z podstawowych kryteriów oceny języków symulacyjnych jest ich efektywność obliczeniowa. Na czas i dokładność obliczeń symulacyjnych istotny wpływ mają: metoda całkowania oraz sposób wyboru kroku całkowania. Problemy całkowania numerycznego są bardzo złożone, liczba metod całkowania jest duża, jednak nie wszystkie nadają się do celów modelowania. Występuje tu niezwykle ważny problem konfliktu między efektywnością i dokład-

nością obliczeń. W związku z tym w językach symulacji ciągłej używa się zaledwie kilku metod całkowania, zapewniających odpowiedni kompromis. Są to np. metody: trapezów, prostokątów oraz Runge-Kutta rzędu drugiego i czwartego.

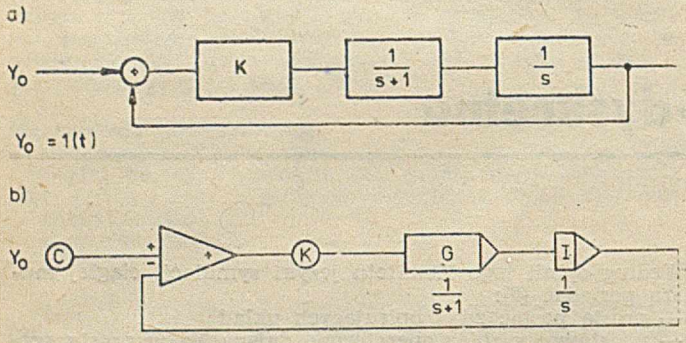
Tabela 1. Język symulacji ciągłej

Język	Klasa języka
CSDP — implementacja na komputerze ODRA-1204 CEMMA — ODRA-1204, K-202 AMF1 — ODRA-1204	Podobieństwo do modelowania analogowego, blokowa struktura operacji matematycznych
MIMIC — CYBER CSSL — CYBER CSMP — IBM, RIAD	Interpretacja wyrażeń matematycznych

CHARAKTERYSTYKA WYBRANYCH JĘZYKÓW SYMULACJI CIĄGŁEJ

CSDP

Struktura języka CSDP [5] jest reprezentatywna dla języków wczesnego rozwoju cyfrowych technik symulacyjnych. Programy pisane w CSDP odpowiadają rozwiązaniom stosowanym w modelowaniu analogowym. Zestaw instrukcji realizuje standardowe operacje, takie jak np.: całkowanie, dodawanie, mnożenie, symulacja pętli histerezy, obszaru martwego, ogranicznika, funkcji skokowej, sinusoidy. Model realizowany w CSDP ma strukturę blokową. Opis problemu modelowania ma poważne wady wynikające z pozycyjnego, stałego formatu danych, braku możliwości posługiwania się nazwami symbolicznymi sygnałów, ograniczonej liczby sygnałów wejściowych itd.



Schemat blokowy (a) i symulacyjny (b) serwomechanizmu

Sposób opisu układu dynamicznego (p. rys.) ilustruje poniższy przykład:

```
1  +  5  —4
2  K  1
3  G  2
4  I  3
5  C

2  1.
5  1.
3  L  1.
   M  1.   1.
```

Kolejne pozycje w każdym wierszu opisu modelu oznaczają numer bloku, typ bloku (I — element całkujący, + — sumator, K — wzmocnienie, G — transmitancja, C — generator stałej) oraz ciąg numerów bloków wejściowych. W czterech ostatnich wierszach zdefiniowane są wartości parametrów bloków operacyjnych. Tak więc dla bloku nr 2 (wzmocnienie) ustalana jest wartość wzmocnienia równa 1, dla generatora stałej (nr 5) ustalana jest wartość skoku jednostkowego. Kolejne dwa wiersze definiują postać transmitancji bloku nr 3, podając wartości współczynników w liczniku (L) i mianowniku (M) członu transmitancyjnego.

CEMMA

Podobnie jak w CSDP, programy w języku CEMMA [4, 6, 8] są przygotowywane na podstawie odpowiednich schematów blokowych. Schematy są opisywane za pomocą zdań strukturalnych języka i wprowadzane do maszyny w postaci programu. Kolejność obliczeń ustalana jest przez translator języka na podstawie odpowiedniego algorytmu sortowania, zapewniającego właściwą sekwencję obliczeniową. Parametry funkcji realizowanych przez człony schematu blokowego mogą mieć nazwy symboliczne.

Fragment programu odpowiadający schematowi rozwiązania równania różniczkowego:

$$\frac{d^2x}{dt^2} + \frac{dx}{dt} + x = 0$$

z warunkami początkowymi:

$$x\Big|_{t=0} = -10{\cdot}5 \qquad \frac{dx}{dt}\Big|_{t=0} = 0$$

można zapisać następująco:

```
1  DOD  3 =—1  2 =—1;
2  INT  1;
3  INT  2 =—10.5;
```

W pierwszym wierszu opisany jest operator sumy ważonej **DOD**, sumujący z wagami równymi —1 sygnały wyjściowe z bloków opisanych w drugim i trzecim wierszu (wartości wag zapisywane są po znakach równości). Następnie opisane są operatory całkowania **INT**, generujące sygnały **dx/dt** oraz **x**, dla sygnału **x** zdefiniowana została wartość początkowa (wartość —10,5 po znaku równości).

Ważną cechą języka CEMMA jest możliwość definiowania makrooperacji składających się z ciągu zdań strukturalnych, opisujących grupę elementów standardowych tworzących dany blok. Raz zdefiniowany makroblok może być używany wielokrotnie do opisu danego układu.

AMF1

Język AMF1 [6] ze względu na metodykę programowania stanowi cyfrowy symulator maszyn analogowych. Zachowuje on takie zalety programowania analogowego, jak: łatwość przygotowywania programów bez koniecznej znajomości metod numerycznych oraz prosta interpretacja fizyczna wielkości występujących w programie. Liczba bloków struktury języka jest mniejsza niż w języku CEMMA. Ważną cechą odróżniającą ten język od poprzednich jest obecność bloku różniczkującego, co znacznie ułatwia rozwiązywanie równań różniczkowych dowolnego typu. Sposób opisu układów dynamicznych jest podobny do opisu stosowanego w języku CEMMA. Tak więc pojedynczy element operacyjny definiowany jest przez podanie numeru porządkowego elementu, kodu literowego elementu, listy numerów bloków wejściowych oraz listy parametrów.

MIMIC

Język symulacyjny MIMIC [1, 2] jest rozszerzeniem symulatora MIDAS. MIMIC stosuje się w modelowaniu układów dynamicznych, opisywanych głównie równaniami różniczkowymi zwyczajnymi oraz wybranymi równaniami cząstkowymi. Język ten wyposażony jest w interpreter wyrażeń matematycznych (dopuszczalne są tu wyrażenia proste połączone operatorami arytmetycznymi, funkcje oraz wyrażenia zawierające przecinki i nawiasy).

Programowanie w języku MIMIC ilustruje przedstawiony poniżej przykład obliczenia całki:

```
CON (A,PI)
PAR (W,FI)
INT  (SIN((W*T+FI), 0.)*A
```

```
FIN (T,PI)
OUT (T,Y)
END
```

Deklaracja **CON** informuje, że **A** oraz **PI** są stałymi o zdefiniowanych przez użytkownika wartościach numerycznych, zaś deklaracja **PAR** — że nazwy **W** i **FI** odpowiadają takim parametrom, których wartości mogą być zmieniane w tym samym przebiegu symulacyjnym. Okres całkowania jest równy **T**, a wartości kątów podawane są w radianach. Deklaracja **OUT** określa, że użytkownika interesuje przebieg funkcji **Y** w czasie od **0** do **T**.

W programie instrukcje nie muszą być pisane w sekwencji wykonania (są wyjątki). Procesor MIMICA przy kompilacji sam ustala odpowiedni porządek wykonywania rozkazów programu. MIMIC zapewnia możliwość wprowadzania podprogramów FORTRANU oraz umożliwia definiowanie podprogramów własnych. W MIMICU można programować funkcje uwikłane i wprowadzać funkcje tabelaryzowane.

CSSL

Język CSSL [3, 6] nazywany językiem wzorcowym modelowania układów ciągłych, realizuje kolejny etap zbliżenia struktury programu symulacyjnego do naturalnej postaci równań stanu. Cecha ta jest bardzo istotna, bowiem interpretacja modelowanych procesów w kategoriach grafu przepływu sygnałów traci swe walory w zastosowaniu do układów mechanicznych lub w zastosowaniach pozatechnicznych.

W języku CSSL wyróżnia się następujące instrukcje:

- instrukcje działania — instrukcje języka algorytmicznego związanego z CSSL (ALGOL, FORTRAN, PL/I), realizujące te wszystkie obliczenia, które wspomagają proces właściwego modelowania
- instrukcje sterujące — podstawienia, definicje, sterowanie diagnostyką
- operatory modelowania (zbiór operatorów języka CSSL można rozszerzyć za pomocą makro-pseudobloków).

Ważną cechą języka CSSL jest zastosowanie definicji przez pominięcie, tzn. w przypadku gdy programista nie dokonał wyboru jednej z dopuszczalnych możliwości — translator wybierze najbardziej prawdopodobną.

CSSL jest językiem o wyraźnie rozbudowanej strukturze i dużych możliwościach aplikacyjnych. Segmentowa struktura programów, duży zbiór instrukcji sterujących, korzystanie z tzw. ograniczników CSSL (np. ARRAY, DATA, DELAY, END, GO, LIMIT, DEFINE, PRINT) znacznie ułatwiają pisanie programów symulacyjnych.

CSMP

Język CSMP [6, 8, 10] jest przykładem wysoko rozwiniętego języka symulacyjnego. Powstał on w wyniku rozbudowy symulacyjnego języka DSL. CSMP związany jest z FORTRANEM oraz ma wszystkie zalety języka zorientowanego blokowo. W języku CSMP występują trzy rodzaje instrukcji:

- instrukcje strukturalne — instrukcje i funkcje standardowe FORTRANU oraz bloki funkcjonalne CSMP
- instrukcje opisu danych — deklaracje danych
- instrukcje sterujące — określają sposób wykonywania programu i określają postać informacji wyjściowej.

Ważną cechą CSMP jest sekwencyjność. Z pewnymi wyjątkami instrukcje strukturalne mogą być pisane w dowolnej kolejności i zgodnie z życzeniem użytkownika — automatycznie sortowane. Podobnie jak w CSSL, w języku CSMP istnieją możliwości definiowania funkcji przy wykorzystaniu instrukcji sterujących MACRO i PROCEDURE lub podprogramów FORTRANU. Struktura programu i instrukcje CSMP są podobne do struktury programów w CSSL.

Przedstawiony niżej ciąg instrukcji bloku MACRO definiuje funkcję opóźniającą z warunkami początkowymi różnymi od zera:

```
MEMRY ZWLOKA
MACRO WY=ZWLOKA (IC,N,PAR,WE)
PROCEDURAL
    IF (TIME.EQ.0.0)WY=IC
      X=DELAY (N,PAR,WE)
    IF (TIME.GT.PAR)WY=X
ENDMAC
```

Zestawienie cech języków. Metody całkowania

Jak wspomniano już w zagadnieniach symulacji ciągłej jednym z najważniejszych zagadnień jest efektywność i dokładność obliczeń. Konieczność zadawania kroku całkowania, wymagana w niektórych językach, może być przyczyną wielu błędów, zwłaszcza gdy użytkownikowi obce są problemy numeryczne.

Tabela 2. Metody całkowania stosowane w językach symulacji ciągłej

Język	Metoda całkowania	Krok całkowania
CSDP	Runge-Kutty drugiego rzędu	Zadawany
CEMMA	Prostokątów, trapezów, Adamsa trzeciego rzędu, Simpsona, Milne'a, Runge-Kutty ze zmiennym krokiem	Zadawany; w metodzie ze zmiennym krokiem zadawany jest tylko dopuszczalny bezwzględny błąd całkowania — krok dobierany jest automatycznie
AMF-1	Eulera	Zadawany
MIMIC	Runge-Kutty czwartego rzędu ze zmiennym krokiem	O ile nie jest zadany, zmieniany automatycznie tak, aby błąd względny nie przekroczył $5 \cdot 10^{-6}$
CSSL	Eulera, Runge-Kutty czwartego rzędu, predykcyjna Adamsa, predykcyjno-korekcyjna Adamsa	Zadawany albo definiowany przez pominięcie (program sam ustala wartość najbardziej prawdopodobną)
CSMP	Piątego rzędu Milne'a, Runge-Kutty czwartego rzędu, Adamsa drugiego rzędu, trapezów, prostokątów	Zadawany; w metodach Milne'a oraz Runge-Kutty zaś tak dobierany automatycznie, aby nie przekroczył ustalonego błędu

W tabeli 2 podano, jakie metody całkowania stosowane są w poszczególnych językach oraz jak dobierany jest krok całkowania.

Dostosowanie metody całkowania do specyfiki problemu pozwala osiągnąć większą precyzję obliczeń, dlatego też bardziej cenione są języki oferujące szerszy aparat obliczeń numerycznych (CEMMA, CSSL, CSMP).

Ważną cechą jest łatwość zapisu i przejrzystość programów symulacyjnych. W tabeli 3 zebrano istotne, z użytkowego punktu widzenia, cechy omówionych języków symulacyjnych.

Tabela 3. Charakterystyka języków symulacji ciągłej

Język	Cechy charakterystyczne
CSDP	Pozycyjny format zapisu, nie ma nazw symbolicznych, sygnałów i parametrów
CEMMA	Pozycyjny format zapisu, dopuszczalne są nazwy symboliczne. Makrooperacje
AMF-1	Pozycyjny format zapisu, nazwy symboliczne, możliwość używania operatora różniczkowania
MIMIC	Wyrażenia matematyczne mogą być parametrami bloków. Makropseudobloki
CSSL	Wyrażenia matematyczne mogą być parametrami bloków. Podział na instrukcje działania, sterujące oraz operatory. Makropseudobloki. Definicje przez pominięcie
CSMP	Wyrażenia matematyczne mogą być parametrami bloków. Podział na instrukcje działania, sterujące oraz operatory. Makropseudobloki. Cechy języka zorientowanego blokowo

Omówione języki nie wyczerpują całego zbioru języków i pakietów do symulacji ciągłej stosowanych w naszym kraju. Celem autorki było tutaj pokazanie kierunku rozwoju techniki cyfrowej symulacji ciągłej, na przykładzie języków najbardziej rozpowszechnionych z jednoczesnym podkreśleniem ich charakterystycznych cech. W przeglądzie pominięto — na przykład — język interpretacji wyrażeń matematycznych GODYS, czy też takie specjalizowane systemy, jak: system analizy obwodów elektrycznych ECAP [6] czy też nowoczesny w swej koncepcji język SIVAL-1204 [6].

PROGRAMOWANIE SYMULACJI CIĄGŁO-DYSKRETNEJ

Oczywiście, nie jest łatwe znalezienie przykładów systemów w pełni ciągłych lub w pełni dyskretnych. Często związki między procesami ciągłymi i dyskretnymi mogą być na tyle silne, że ich pominięcie w analizie problemu i w modelu symulacyjnym jest niedopuszczalne. Problemy tego typu występują w zagadnieniach sterowania i zarządzania produkcją, w problemach sterowania obiektami dynamicznymi, w meteorologii, technice jądrowej, transporcie itd. Koncepcja języków realizujących proces symulacji mieszanej polega na uwzględnieniu w bazie języka możliwości inicjacji, między kolejnymi zdarzeniami, procesu całkowania numerycznego równań stanu.

Baza języka symulacji mieszanej odpowiada językowi symulacji dyskretnej. Do tej bazy dodany jest zespół procedur operacji numerycznych oraz blok realizujący inicjowanie zdarzeń dyskretnych przez procesy ciągłe i uaktywnianie procesów ciągłych w wyniku zdarzeń dyskretnych. Model mieszany (ciągło-dyskretny) jest modelem dokładniejszym niż model ciągły lub dyskretny. Można tu bowiem uwzględnić zmiany własności obiektów, będące efektem zmiany warunków, np. na skutek osiągnięcia wartości granicznej parametrów.

Przykładem języka symulacji mieszanej jest GASP IV [8, 9], zbudowany na bazie FORTRANU. Na etapie symulacji dynamiki obiektu w systemie zdarzeń dyskretnych język ten realizuje metodę następstwa zdarzeń.

Koncepcję interakcji procesów wykorzystano natomiast w języku NEDIS, stanowiącym rozszerzenie SIMULI.

* * *

Przedstawiony przegląd problemów nie wyczerpuje zagadnień związanych z symulacją komputerową. Niezwykle szeroka tematyka związana jest — na przykład — ze statystycznymi aspektami symulacji dyskretnej, zagadnieniami projektowania i weryfikacji metod symulacyjnych, czy też opracowywania wyników eksperymentów symulacyjnych.

Istotną sprawą jest również zagadnienie kosztów. Modelowanie systemu, wykonywanie wielu obserwacji oraz pisanie programów dla maszyny cyfrowej jest bardzo pracochłonne. Programy symulacyjne, podobnie jak większość programów komputerowych, zwykle rozrastają się żywiołowo — zarówno co do wielkości, jak i złożoności. Stosowanie języków symulacyjnych znacznie zmniejsza czasochłonność programowania. Płaci się za to pewnym obniżeniem elastyczności, nieznacznym przy stosowaniu odpowiednio dobranych języków, oraz pewnym pogorszeniem efektywności realizowanego programu.

LITERATURA

[1] Budziaszek J.: Język symulacyjny MIMIC. Skrypt AGH, Kraków, 1979

[2] CDC-600 Computer Systems. MIMIC. Digital Simulation Language. Reference Manual, USA, 1972

[3] CDC-CSSL III Users Guide Version 1. USA, 1974

[4] Findeisen W. (red.): Poradnik inżyniera automatyka. WNT, Warszawa, 1973

[5] Frelek B.: Symulacja w języku CSDP. Wydawnictwo Politechniki Warszawskiej, Warszawa, 1978

[6] Kącki E., Woźniakowski M.: Modelowanie analogowe, hybrydowe oraz cyfrową symulacja maszyn analogowych. PWN, Warszawa 1979

[7] Kochenburger R. J.: Modelowanie układów dynamicznych przy użyciu maszyn matematycznych. WNT, Warszawa, 1975

[8] Perkowski P.: Technika symulacji cyfrowej. WNT, Warszawa, 1980

[9] Pritsker A., Hurst N.: GASP IV a combined continuasdiscrete FORTRAN based simulation language. Simulation 1973, 21, 3 s. 65

[10] Ziętara G.: CSMP. Język symulacji układów ciągłych. Opis i instrukcja. Opracowanie ZOWAR, Warszawa, 1973.

ZYGMUNT RYZNAR

ZETO Kraków

Pakiet dokumentowania i wspomagania prac programistycznych PROSTR

Pakiet PROSTR jest produktem ZETO Kraków, opracowanym dla komputerów Jednolitego Systemu z wykorzystaniem systemu operacyjnego OS. Poza komputerową dokumentacją programów pakiet ma właściwości generacyjne, oparte na własnym, językowo niezależnym generatorze oraz zapewnia przeszukiwanie katalogów pakietu.

Pakiet PROSTR jest narzędziem programisty dostosowanym do warunków przemysłowej produkcji oprogramowania. Rdzeniem pakietu jest specjalizowana baza danych, zawierająca elementy konstrukcyjne systemów powtarzalnych wraz z opisem dokumentacyjnym. Elementami mogą być procedury i programy napisane w sposób zwykły albo szkieletowy — wymagający zastosowania preprocesora poprzedzającego kompilację. Preprocesor ten łączy niektóre funkcje edytora i uniwersalnego makrojęzyka. Opis dokumentacyjny zasobów programowych jest sformalizowany i zawiera ok. stu zwrotów specyfikacyjnych, z których część może być opcjonalna w zależności od życzeń użytkownika.

CHARAKTERYSTYKA UŻYTKOWA

Selektywne przeszukiwanie opisu dokumentacyjnego odbywa się za pomocą prostego języka zapytań, nie wymagającego od użytkownika znajomości organizacji wewnętrznej bazy danych. Pakiet PROSTR stwarza więc możliwości łatwego dostępu do istniejących w ośrodku zasobów programowych i ich wykorzystywania bez współudziału autorów. W celu przeciwdziałania ewentualnym nadużyciom, dostęp do pakietu i poszczególnych jednostek katalogowanych może być chroniony hasłami.

Opisy dokumentacyjne mogą dotyczyć zasobów programowych napisanych w dowolnych językach i na dowolne typy komputerów. Poza opisami może być również katalogowany dowolny tekst opatrzony identyfikatorem użytkownika. Stwarza to możliwości użycia pakietu do tworzenia scentralizowanych katalogów oprogramowania, z równoczesnym tworzeniem selektywnych biuletynów informacyjnych i generowaniem oprogramowania w odpowiedniej postaci. Usługi takie pełniłyby funkcję swoistej giełdy oprogramowania.

Generacyjne właściwości pakietu mogą być wykorzystane do działań na dowolnym tekście, np. na zdaniach sterujących do wariantowej generacji taśm dystrybucyjnych, czy po prostu — do uzyskania zdań sterujących do programów. Dokumentacyjne właściwości mogą natomiast stać się podstawą budowy uniwersalnych baz danych formatowo-tekstowych z językiem specyfikacyjnym definiowanym przez użytkowników, gdyż zastosowane w pakiecie sterowane tablicami analizatory języków umożliwiają takie rozwiązanie.

Pakiet PROSTR realizuje następujące funkcje użytkowe:

- katalogowanie opisu zasobów programowych
- katalogowanie tekstów programowych zwykłych i szkieletowych
- generowanie i modyfikację tekstów szkieletowych
- selektywne przeszukiwanie opisu.

Pakiet PROSTR dopuszcza katalogowanie opisu bez tekstu programu, natomiast odwrotna sytuacja nie jest akceptowana.

Szkieletowe teksty programowe mogą być pisane w języku PL/1, COBOL i ASSEMBLER. Inne teksty językowe (np. FORTRANOWE, zdania JCL) również mogą być modyfikowane przy niewielkich ograniczeniach formalnych. Działający na tekstach szkieletowych generator cechuje — w porównaniu z makroaparatem stosowanym w poszczególnych językach programowania — niezależność językowa i prostota. Tekst szkieletowy jest odpowiednikiem szeroko rozumianej makrodefinicji stosowanej na poziomie programu. Umożliwia ona specyfikację wariantów logicznych (poprzez modyfikatory # IF odpowiadające instrukcjom kompilacji warunkowej) oraz użycie instrukcji replikacyjnych (modyfikatory # DO) i parametrów formalnych. Opis modyfikatorów znajduje się poza tekstem szkieletowym i obejmuje jego wartości domyślne, typ operacji, typ wartości oraz zależności w stosunku do innych modyfikatorów.

Ponadto opisywane jest przeznaczenie modyfikatora oraz podawana lista wartości, jakie może przyjmować. Dzięki opisowi unika się szczegółowej analizy tekstu szkieletowego przed jego modyfikacją (instrukcja MODIFY), która polega albo na wstawieniu wartości (lub wyrażenia), albo powieleniu fragmentu programu (ze zmianą nazw zmiennych), albo też na zmianie struktury logicznej programu — poprzez usunięcie zbędnych ścieżek logicznych.

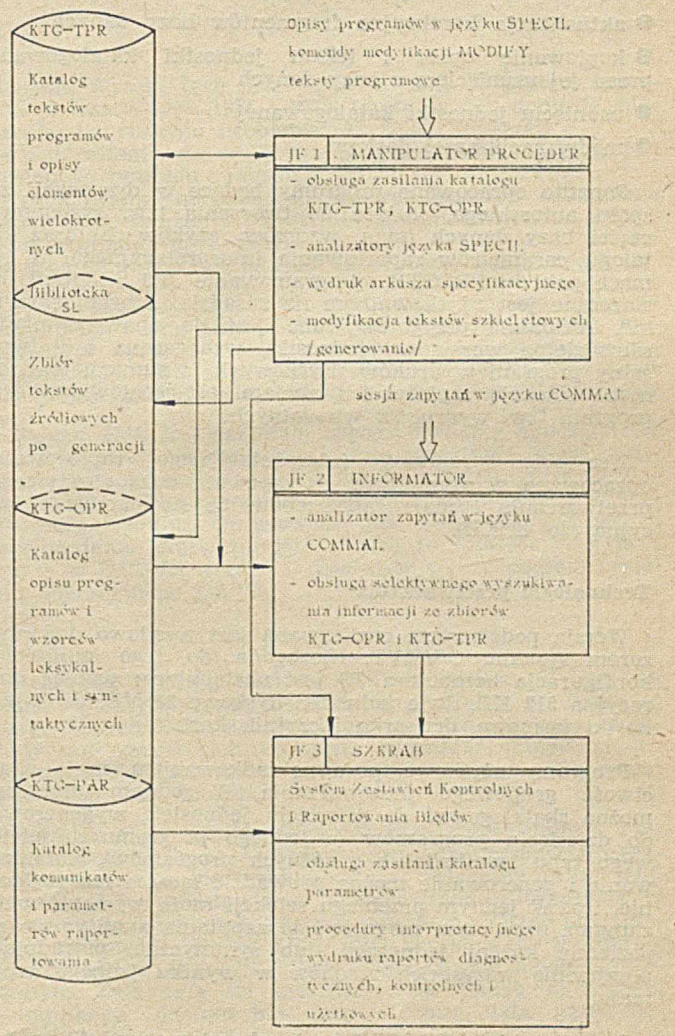
Selektywne przeszukiwanie obejmuje przejrzenie zasobów według kryterium czynnościowego (np. wyszukanie procedury kontroli danych, konwersji kodów, wydruku), tematyki zastosowań (płace, gospodarka materiałowa itp.), dowolnych zwrotów opisowych języka specyfikacyjnego (np. wg języka programowania, wersji kompilatora, komputera, systemu operacyjnego, autora). Opis, generowanie, modyfikacja i selekcja tekstów szkieletowych zostały sformalizowane w postaci nieproceduralnych języków SPECIL i COMMAL.

Efekty stosowania pakietu mają zarówno charakter ekonomiczny, jak i jakościowy. Efekty wymierne uzyskuje się dzięki wykorzystaniu tych samych procedur i programów w różnych systemach oraz z tytułu zmniejszenia pracochłonności dokumentacji programowej. Ponadto uzyskana dokumentacja ma jednolitą, niezależną od wykonawcy formę i jest kompletna. Cechy te zapewnia zastosowanie automatycznej weryfikacji dostarczonej dokumentacji.

Pakiet ulepsza warsztat programisty oraz ułatwia kierowanie pracami programistycznymi, bowiem kierownictwo i wykonawcy uzyskują dokładną orientację w zgromadzonym dorobku i mogą przeciwdziałać dublowaniu prac.

CHARAKTERYSTYKA KONSTRUKCYJNA

Architektura podstawowa pakietu została przedstawiona na rysunku 1. W skład pakietu wchodzą trzy jednostki funkcjonalne, z których każda wyróżnia się nie tylko innym zakresem działania, lecz również specyficznym podejściem technologicznym. Użytkownik ma bezpośredni kontakt z dwoma pierwszymi jednostkami (JF1 i JF2) poprzez język specyfikacyjny SPECIL i język komend COMMAL. W związku z powyższym znaczną część składową tych jednostek stanowią analizatory języków. W przypadku języka SPECIL poza analizatorami tradycyjnymi (leksykalnym i syntaktycznym) występuje również analizator opisu modyfikatorów. JF3 SZKRAB pełni funkcje usługowe w stosunku do dwóch pierwszych jednostek, realizując interpretacyjnie raporty wydruków. Interpretacja polega na tym, że podczas wykonywania programów ustalany jest układ stron, struktura nagłówka, struktura wierszy danych, zależnie od parametrów przekazywanych w wywołaniu procedur raportowania. Zmieniając wartości parametrów w zbiorze KTG-PAR, można bez zmian programowych we wszystkich tabulogramach zmienić np. nazwy użytkowników, ośrodka, brzmienie nagłówków, komunikatów, formaty stron itp., co ma znaczenie przy rozpowszechnianiu pakietu PROSTR. Zawartość zbioru parametrów ustalana jest przez zespół autorski.



Rys. 1. Architektura pakietu PROSTR

Baza danych

Baza danych składa się z dwóch wzajemnie powiązanych zbiorów KTG-OPR i KTG-TPR, zawierających zarówno tablice wzorców językowych, jak i informacje opisowe oraz teksty programowe. Zbiór KTG-OPR ma organizację indeksowo-sekwencyjną, zaś zbiór KTG-TPR — organizację członowo-sekwencyjną. W każdym ze zbiorów wystę-

puje wiele typów zapisów, z reguły składających się ze zmiennej liczby 80-bajtowych segmentów. Umożliwia to katalogowanie tekstów zmiennej długości. Powiązania pomiędzy typami zapisów oraz organizacje zbiorów są niewidoczne dla użytkownika, gdyż odwołuje się on do danych w sposób nieproceduralny, nie wnikając w to, w jakich znajdują się one zbiorach i zapisach.

W celu zapobieżenia zbyt częstym reorganizacjom, zbiór KTG-OPR może być specjalnie sformatowany przez generowanie obszaru użytkowego według życzeń użytkowników. W celu przyspieszenia przeszukiwania zbioru KTG-TPR zapewniono bezpośredni dostęp do jego sekcji, gromadzących opisy modyfikatorów, zbiorów, struktur danych, pętli itp. Dostęp do danych może być chroniony przez hasła.

Oprogramowanie

Oprogramowanie narzędziowe pakietu składa się z 20 programów, napisanych w językach PL/1 i ASSEMBLER. Obsługę użytkowników-programistów realizują trzy programy, które w wersji nakładkowej wymagają 180—200 KB pamięci operacyjnej. Większość programów przeznaczona jest dla administratora bazy danych i służy do wykonywania takich zadań, jak:

- wydruki arkuszy specyfikacyjnych dla programistów
- aktualizacja korekcyjna fragmentów bazy danych
- kopiowanie opisu i tekstu jednostki katalogowanej przed jej usunięciem z bazy danych
- usunięciu jednostki katalogowanej
- modyfikacji haseł dostępu.

Ponadto opracowano programy będące w dyspozycji zespołu autorskiego, służące do tworzenia tzw. systemowej części bazy danych (m.in. wzorców językowych) oraz katalogu parametrów raportowania interpretacyjnego. W ramach pakietu PROSTR oprogramowanie JF3 SZKRAB dostarczane jest użytkownikom na zasadzie „czarnej skrzynki". Aktualnie przygotowywana jest do upowszechniania samodzielna wersja tej jednostki, wzbogacona o uniwersalny program wydruków użytkowych (z sumami kontrolnymi, selekcją zapisów i funkcjami obliczeniowymi) oraz program tzw. wydruków wirtualnych.

Wszystkie programy interpretacyjnego raportowania opracowano w asemblerze, co zapewnia dużą efektywność przetwarzania i małe zapotrzebowanie na pamięć operacyjną (do 40 KB).

Technologia przetwarzania

Wersja podstawowa realizowana jest wsadowo pod nadzorem systemu OS/MVT. Niezbędna do tego minimalna konfiguracja komputera JS jest następująca: pamięć operacyjna 512 KB, dwie jednostki dyskowe 30 MB, dwie jednostki taśmowe, drukarka i czytnik kart.

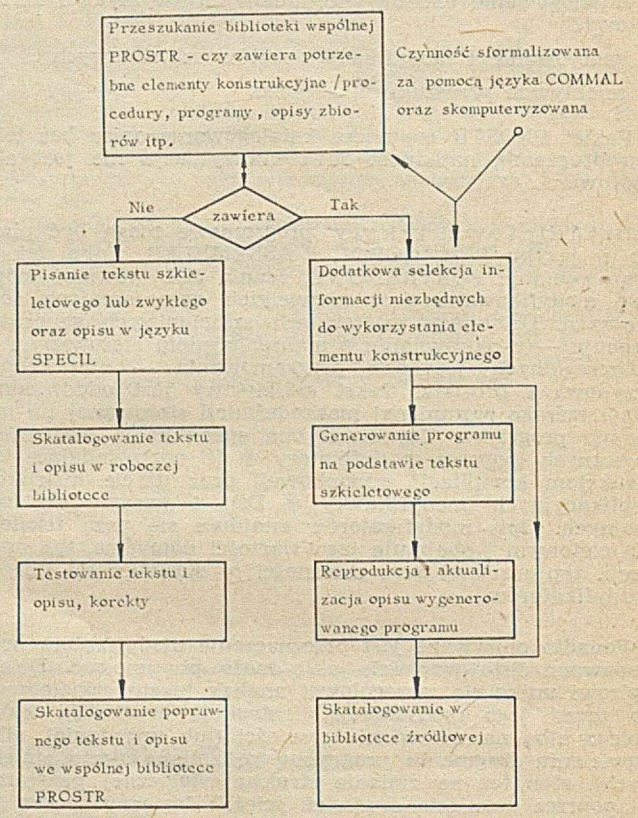
Programy użytkowe, poza nakładkowaniem, mają właściwość grupowego przetwarzania. W jednym przebiegu można skatalogować do dziesięciu jednostek, wygenerować do dziesięciu programów (z jednego programu szkieletowego typu „matka" lub z różnych programów), katalogowanie i generowanie można prowadzić łącznie lub rozłącznie, itp. W jednym przebiegu selekcji może wystąpić wiele zapytań, przy czym mogą być to zapytania analityczne (na poziomie każdej jednostki) lub syntetyczne, powodujące przejrzenie wszystkich zasobów w wyniku jednego zapytania.

Języki pakietu PROSTR mają charakter modyfikowalny i w pewnym zakresie mogą być dostosowywane do potrzeb poszczególnych użytkowników.

NARZĘDZIA

Przejście z tradycyjnej techniki programowania na programowanie szkieletowe jest nie tylko zabiegiem technicznym, lecz również metodycznym, przesuwającym ciężar prac z kodowania programu na jego zaprojektowanie. Problemy projektowe dotyczą m.in. typizacji algorytmu (aby mógł być używany w różnych zastosowaniach) oraz podziału strukturalnego na bloki proceduralne — pod kątem podatności na włączanie lub wyłączanie z wariantów programu. Kodowanie programu szkieletowego może początkowo nastręczać trudności (m.in. techniczne i psychologiczne), gdyż poza językiem programowania programista musi znać reguły stosowania modyfikatorów i pisać program „dwutorowo", nakładając na tekst źródłowy instrukcje sterujące modyfikacją programu.

Ogólny schemat postępowania programisty przedstawiono na rysunku 2. Przed przekazaniem programu szkieletowego do wspólnej biblioteki programista testuje go we własnej bibliotece roboczej, korzystając z instrukcji dokumentacji użytkowej i z pomocy administratora pakietu, który m.in. pełni funkcje konsultacyjne. Opis programu nanoszony jest na komputerowo drukowane arkusze specyfikacyjne, których zakres i objętość można parametryzować. W wersji rozwiniętej przewiduje się możliwość tworzenia opisu na monitorach ekranowych oraz generowania dokumentacji programowej do programów. Ponadto rozszerzone będą znacznie możliwości generowania programowego (m.in. w zakresie łączności międzymodułowej).



Rys. 2. Ogólny schemat postępowania przy użytkowaniu pakietu

Przed przystąpieniem do pisania programu szkieletowego, programista powinien, używając języka COMMAL, dokonać selektywnego przeszukiwania wspólnej biblioteki w celu sprawdzenia, czy nie ma w niej jednostki, która odpowiadałaby jego wymogom. Po znalezieniu odpowiedniego tekstu szkieletowego, przeszukuje on głębiej opisy wybranych jednostek, sięgając po reguły stosowania poszczególnych modyfikatorów.

* * *

Ze względu na ograniczone ramy artykułu nie omówiono w nim wielu istotnych właściwości pakietu. Pominięto m.in. opis języka SPECIL i COMMAL oraz rozbudowanego systemu zabezpieczenia hasłowego i możliwości programów administratora. Nie poruszono również spraw organizacyjnych i metodycznych.

Na zakończenie należy podkreślić pewne uwarunkowania, których spełnienie powinno ułatwić założenie biblioteki oprogramowania powtarzalnego, która stanowi o celowości zastosowania pakietu PROSTR.

Wydaje się, że szkieletowe jednostki katalogowe powinny odpowiadać następującym wymaganiom:
— zawierać modyfikowalny algorytm przetwarzania albo/ /oraz
— zawierać zmienne wejścia (ten sam algorytm stosowany na różnych zbiorach) albo/oraz
— być programem wielofunkcyjnym, z którego dla pojedynczych zastosowań wybiera się poszczególne funkcje, albo
— być programem zbiorczym typu „matka", zawierającym wiele elementów nadmiarowych (np. opisy wszystkich zbiorów systemu oraz procedury ich czytania i aktualizacji), które wymagają pominięcia w przypadku poszczególnych programów.

Walory generacyjne pakietu PROSTR mogą być wykorzystane nie tylko w przypadku systemów powtarzalnych, lecz również indywidualnych, w których zachodzą częste zmiany algorytmów (np. w przypadku systemów płacowych).

Historia Systemów Zarządzania Bazami Danych

W wyniku prac organizacji CODASYL (ang. COnference of DAta SYstems Languages) powstaje język programowania COBOL, którego dział opisujący przetwarzane dane (ang. Data Division) stanowi pierwszy krok ku wyodrębnieniu definicji danych.
Pojawiają się zwiastuny przyszłych komercyjnych generatorów programów użytkowych (np. MARK IV firmy INFORMATICS INCORP., Report Program Generator — RPG II).
Agendy wojskowa i wywiadowcza rządu Stanów Zjednoczonych popierają pierwsze systemy zarządzania danymi, które jednak przeważnie nie spełniają pokładanych w nich nadziei.

1960

Komercyjne przetwarzanie danych rozwija się w sposób bezprecedensowy. Wyrazem tego rozwoju jest:
- oddanie do powszechnego użytku języka programowania COBOL
- dostępność na rynku dużych komputerów
- badania wielu firm w dziedzinie techniki zarządzania danymi
- eksperymenty z systemami przetwarzania w trybie bezpośrednim (ang. on-line).

Na czoło zainteresowań wielu naukowców wysuwają się zagadnienia związane z tworzeniem zbiorów głównych oraz zastosowaniem przetwarzania partiowego (wsadowego).

1964

Charles Bachman tworzy pierwszy dostępny na rynku System Zarządzania Bazą Danych (SZBD) o nazwie IDS (ang. Integrated Data Store). Pierwszą wersję tego systemu, zrealizowaną przez firmę GENERAL ELECTRIC (obecnie HONEYWELL), przeznaczono do wykorzystywania na maszynie GE-200. Charles Bachman, jeden z członków założycielskiej Grupy Roboczej CODASYLU ds. Baz Danych, uczestniczył w pracach tej grupy w latach 1966—1968. Jego zasługą jest to, że ustalone przez tę grupę wymagania techniczne są realizowane przez wielu dostawców oprogramowania w różnych częściach świata. Choć IDS powstał o kilka lat wcześniej niż wymagania techniczne CODASYLU, może on uchodzić za realizację tych wymagań. Na systemie IDS opiera się bezpośrednio wiele komercyjnych pakietów oprogramowania.

1968

Następuje olbrzymi postęp w dziedzinie technologii baz danych. Organizacja CODASYL rozpoczyna intensywne prace w zakresie technologii baz danych. Jej zainteresowania obejmują m.in. problemy standaryzacji języków baz danych. Wyniki tych prac opublikowano w postaci raportów w latach 1969, 1971 i 1973. Uzyskały one doniosłe znaczenie dla rozwoju problematyki SZBD, poprzez:
— podkreślenie, że w języku programowania COBOL jest dostęp do poziomu rekordu
— promowanie niezależności danych poprzez oddzielenie problemów związanych z definiowaniem danych od zagadnień manipulowania danymi
— uznanie integracji danych za problem podstawowy.

Prace projektowane nad udoskonaleniem opracowywanego przez firmę IBM SZBD IMS (ang. Information Management System) zostają zatrzymane ze względu na zaangażowanie IMS w program kosmiczny APOLLO, którego celem jest lądowanie człowieka na Księżycu w 1969 roku.

System IMS jest pakietem oprogramowania maszyn IBM, który w swej podstawowej formie umożliwia pracę jedynie w trybie wsadowym. System ten można jedak rozbudować do obsługi transmisji danych i zastosowania w trybie bezpośrednim. Opracowano już wiele różnych wersji tego systemu. Najważniejsze z nich to IMS/360 (wersja 1 i 2) oraz IMS/VS wersja 1, działające pod nadzorem systemu operacyjnego OS/VS.

1970

Związany z firmą IBM E. F. Codd publikuje rozprawę naukową zatytułowaną „A Relational Model of Data for Large Shared Data Banks", która zawiera pierwszy matematyczny sposób potraktowania zagadnień danych i ich manipulacji oraz stanowi podstawę teoretyczną dla nowej technologii baz danych.

Rozwija się i utrzymuje przez całe dziesięciolecie rozłam w poglądach na tę sprawę pomiędzy przedstawicielami przemysłu a teoretykami. Model relacyjny prezentacji danych jest przedstawiany jako poważna alternatywa w stosunku do odpowiadającego standardom CODASYLU modelu sieciowego. Firma IBM jest przekonana, że dalszy rozwój będzie dotyczył SZBD opartych na modelu relacyjnym. Zaczynają się spekulacje na temat możliwości i implikacji „maszyn baz danych" (ang. data base machine).

1970—1971

Pierwsze próby wprowadzenia SZBD do powszechnego użytku.

Większość dostawców sprzętu (w tym również firma IBM) oferuje SZBD dostosowane do współpracy z różnymi systemami operacyjnymi.

Model sieciowy, odpowiadający standardom CODASYLU, jest preferowany przez dostawców sprzętu (z wyjątkiem

firmy IBM). Najbardziej znane SZBD reprezentujące podejście sieciowe to:
— IDS/1 firmy HONEYWELL
— DMS 1100 (Data Management System) firmy UNIVAC
— EDMS firmy XEROX.

Firma BURROUGHS angażuje się w produkcję SZBD o nazwie DMS II, z którego to zamierzenia później rezygnuje.

Pojawiają się dostawcy SZBD produkowanych na rynek i przeznaczonych do wykorzystywania na sprzęcie firmy IBM. Dostawcy ci (CULLINANE, MRI, CINCOM SYSTEMS) oferują systemy konkurencyjne w stosunku do IMS. Inną interesującą implementacją raportu grupy DBTG CODASYLU jest IDMS (Integrated Database Management System), sprzedawany przez firmę CULLINANE i wykorzystywany przede wszystkim na komputerach firmy IBM. Firma MRI produkuje i wypuszcza na rynek SYSTEM 2000, natomiast firma CINCOM SYSTEMS — SZBD TOTAL.

Wkrótce pojawiają się następni dostawcy SZBD dla sprzętu firmy IBM, tacy jak:
— firma COMPUTER CORPORATION OF AMERICA — C.C.A (MODEL 204)
— SOFTWARE A.G (ADABAS — Adaptable DAta BAse System)
— firma INFODATA (INQUIRE)
— firma APPLIED DATA RESEARCH — ADR (DATACUM/DB).

Wszystkie ww. systemy wykorzystują listy inwersji danych oraz języki zapytań (języki bezpośredniego użytkownika).

Dla użytkowników wykorzystujących sprzęt firmy IBM momentem decydującym dla wprowadzenia technologii baz danych staje się (i pozostaje przez całe dziesięciolecie) wybór odpowiedniego SZBD.

Wszystkie dostępne na rynku pakiety oprogramowania mają na celu wspomaganie bezpośredniego dostępu do baz danych.

1973

Amerykańskie stowarzyszenie informatyczne ACM przyznaje twórcy i architektowi IDS, pierwszego dostępnego na rynku SZBD Charlesowi Bachmanowi, zaszczytną Nagrodę Turinga za wybitne osiągnięcia w dziedzinie baz danych. Odczyt Bachmana wygłoszony z tej okazji zatytułowany „The Programmer as Nawigator" konfrontuje wcześniejszy pogląd na przetwarzane dane (w myśl którego elementem centralnym był komputer, a dane traktowano jako strumień płynący przez maszynę w trakcie ich przetwarzania) z poglądem nowoczesnym, w myśl którego baza danych jest podstawowym zasobem, a komputer tylko narzędziem umożliwiającym dostęp do niej. Terminu n a w i g a c j a Bachman użył dla określenia procesu poruszania się w bazie danych po wyraźnie określonych ścieżkach, od jednego rekordu do drugiego, w poszukiwaniu żądanych elementów danych.

W tym samym roku Leo J. Cohen ze współpracownikami publikuje rozprawę naukową „DBMS: A Critical and Comparative Analysies", która staje się „biblią" użytkowników SZBD.

1973—1978

Bazy danych stają się główną częścią technologii zastosowań gospodarczych, a SZBD ukazują swe zalety w różnorodnych środowiskach sprzętowych i programowych. Coraz powszechniej dostrzega się korzyści stosowania technologii baz danych w stosunku do metod tradycyjnych.

Do najważniejszych zalet koncepcji baz danych należą:
— ograniczenie redundancji danych
— usprawnienie dostępu do danych
— zwiększenie integracji danych
— zwiększenie liczby użytkowników tych samych zbiorów danych
— umożliwienie łatwej rozbudowy i modyfikacji istniejących systemów przetwarzania danych
— zmniejszenie kosztów utrzymania zbiorów danych
— zwiększenie możliwości wykorzystywania zbiorów w trybie bezpośrednim.

W przedsiębiorstwach, w których wdrożono systemy baz danych pojawiają się Administratorzy Baz Danych, niestety często bez odpowiedniej pozycji w hierarchii administracji przedsiębiorstwa i z ograniczonym zakresem kompetencji. Podejmuje się wiele przedsięwzięć przyczyniających się do rozwoju standardowych metodologii oraz akcentuje potrzebę ułatwionego dostępu do baz danych, chociaż postęp w tej dziedzinie jest jeszcze niewielki. Powstałe stowarzyszenie użytkowników SZBD domaga się zapewnienia większego wpływu na postęp w dziedzinie rozwoju SZBD. Wszyscy dostawcy SZBD uzyskali rangę znacznych przedsiębiorstw, a na ich czoło wysunęła się firma CULLINANE.

Nie do pomyślenia jest projektowanie SZBD ukierunkowanego tylko na konkretne zastosowanie.

1976

Na rynku coraz częściej i liczniej zaczynają pojawiać się Systemy Słowników Danych (ang. Data Dictionary Systems). Na początku pełnią one funkcję dokumentacji i narzędzia wspomagającego SZBD, są one bowiem w istocie bazami danych zawierającymi „dane o danych". Systemy Słowników Danych będą rozszerzać swe oddziaływanie i możliwości. Sądzi się, że do 1980 roku staną się one istotnym składnikiem funkcji zarządzania danymi oraz pożyteczną pomocą w opracowywaniu systemów baz danych.

Producenci minikomputerów oraz dostawcy oprogramowania oferują mini-SZBD, przeznaczone do wykorzystywania na minikomputerach i coraz bardziej popularne w zdecentralizowanym lub rozłożonym przetwarzaniu danych (ang. distributed data processing).

Cztery lata wcześniej Komitet Planowania i Ustalania Wymagań Standaryzacji (Standards Planning and Requirement Committee — SPRAC) amerykańskiej organizacji normalizacyjnej ANSI (American National Standards Institute) powołał grupę ds. SZBD z zadaniem prowadzenia działań normalizacyjnych oraz opracowywania zaleceń w zakresie problematyki baz danych. Przyjęto, że standaryzowane mogą być jedynie interfejsy, a główny wysiłek skoncentrowano na definiowaniu uogólnionej architektury baz danych oraz odpowiednich interfejsów. W raporcie tej grupy znajduje się szczegółowy opis takiej architektury z uwypukleniem schematu pojęciowego. Raport ten wskazuje na główne kierunki przyszłościowego rozwoju SZBD.

Nasilają się spory rzeczników podejścia sieciowego z obrońcami modelu relacyjnego, a istniejący pomiędzy zwolennikami obu modeli rozłam utrwala się.

ANSI nie jest w stanie posunąć naprzód problemu standaryzacji języków baz danych. Najważniejszym oponentem w tej sprawie jest firma IBM. Powoduje to czasowe usunięcie się z pola działania grupy ANSI/SPRAC.

W laboratoriach naukowych powstają tymczasem opracowania prototypowe oparte na relacyjnym modelu danych. Najciekawsze z nich to: system INGRES (INteractive Graphics and REtrieval System) firmy U.C. (Berkeley) oraz SYSTEM R., projektowany i rozwijany w laboratorium badawczym firmy IBM w San Jose. Ten ostatni jest jeszcze prototypem, nie produktem. W systemie tym użytkownik wykorzystuje język SEQUEL — zarówno do pracy w trybie bezpośrednim, jako samodzielny język konwersacyjny, jak i — za pośrednictwem języka macierzystego (np. PL/1) — jako wsadowy język danych.

Sieci informatyczne umożliwiają użytkownikom nie tylko wspólne korzystanie z mocy obliczeniowej, ale — co ważniejsze — pozwalają na wspólne użytkowanie danych w skali zmieniającej obecną koncepcję baz danych. Technologia baz rozłożonych jest stosunkowo nowa. Polega ona na tym, że zamiast gromadzenia dużej ilości danych w scentralizowanej bazie w jednym ośrodku obliczeniowym, dane są przechowywane w miejscach ich powstawania lub w ich pobliżu. Innymi słowy — dane nie są umieszczone w jednym fizycznym urządzeniu pamięciowym, ale rozłożone w sieci rozproszonych geograficznie komputerów. Na przykład, baza danych osobowych firmy z zakładami położonymi w różnych częściach kraju, może być rozmieszczona w taki sposób, by informacja o danych pracownika była przechowywana w węźle położonym najbliżej zakładu, w którym ten pracownik jest zatrudniony. W tym przypadku można oczekiwać, że koszty użycia łącza (koszty

przekazywania wiadomości) zostaną sprowadzone do minimum, dzięki przechowywaniu danych w tym węźle, z którego pochodzi większość zapytań.

Uzasadnieniem rozłożenia elementów bazy danych między kilka węzłów są także względy bezpieczeństwa i niezawodności dostępu do danych. Chociaż realizacja systemów rozłożonych baz danych nastręcza jeszcze wiele problemów technicznych, koncepcja ta spotyka się z dużym zainteresowaniem.

Dostęp do zbioru oraz bazy danych wymaga wykonania znacznej liczby rozkazów. Jedyną drogą prowadzącą do zmniejszenia obciążenia procesora oraz do zwiększenia w ten sposób jego przepustowości jest przeniesienie do urządzenia sterującego pamięcią całości funkcji związanych z dostępem do zbiorów oraz (lub) z zarządzaniem bazą danych. Realizuje się to kilkoma sposobami:

- firma CULLINANE CORPORATION opracowuje system, w którym zarządzanie bazą danych IDMS zainstalowane zostało w minikomputerze powiązanym z większym systemem komputerowym; minikomputer oraz IDMS mogą działać jako inteligentne urządzenie pamięciowe.
- firma IBM opracowuje inteligentne urządzenie pamięciowe, realizujące rozkazy dostępu do bazy danych w SZBD IMS; ten eksperymentalny podsystem powiązany jest wzajemnie z konwencjonalnym komputerem serii IBM/370
- firma BELL buduje i produkuje procesor dla komputerów serii UNIVAC 1100, który realizuje rozkazy języka manipulowania danymi systemu DMS 1100
- firma IMS ASSOCIATES wychodzi na rynek z systemem opartym na mikroprocesorze, a zaprojektowanym w celu akomodacji oprogramowania SZBD oraz wiązania systemów komputerowych lub inteligentnych terminali.

Pojawia się zainteresowanie maszynami dla SZBD przetwarzania rozłożonego. Prowadzi się intensywne studia w tej dziedzinie oraz konstruuje prototypy takich maszyn.

1979

Potrzeba posiadania Słownika Danych staje się coraz bardziej wyraźna. Definiuje on i kontroluje środowisko danych. W kręgu zainteresowań dostawców oprogramowania pojawia się problematyka funkcji Systemów Słowników Danych (SSD), wspomagającej SZBD. Wiele dostępnych na rynku SSD oferuje infromacje jedynie o bazie danych. Inne są bardziej wszechstronne: kontrolują definicje pól i schematów otoczenia bazy danych, generują raporty opisujące organizację danych itp. Wszystkie zmiany w danych użytkowych muszą mieć swoje odbicie w Słowniku Danych.

Zaczynają pojawiać się generatory programów użytkowych oraz języki wyższego poziomu przeznaczone dla SZBD, a grupa ASNSI/SPRAC ponownie zaznacza swoją obecność, lecz pomimo nacisków ze strony rządu USA oraz zwolenników rozłożonego przetwarzania danych — postęp w zakresie standaryzacji jest niewielki. Zainteresowanie koncepcją SZBD przetwarzania rozłożonego maleje w miarę pojawiania się problemów związanych z ekonomiczną stroną zmian w rozwiązaniach sprzętu oraz łączności telekomunikacyjnej.

Oprócz systemów przeznaczonych na rynek, w uniwersytetach i instytutach naukowych buduje się wiele systemów eksperymentalnych, wśród których jednym z najciekawszych jest SYSTEM R. Prace nad tym pakietem oprogramowania firmy IBM, opartym na modelu relacyjnym, pozostają jednak otoczone tajemnicą.

1980

Zainteresowania koncentrują się wokół problemów związanych z administrowaniem danymi. Wzrasta liczba Administratorów Baz Danych w przedsiębiorstwach. Zajęcie to wymaga głębokiej wiedzy technicznej oraz umiejętności rozumienia i interpretacji zarządzania. Wiele pytań związanych z tymi problemami pozostaje nadal bez odpowiedzi.

Dostawcy SZBD rozszerzają swoją działalność zwłaszcza na takie dziedziny, jak: integracja asortymentu wyrobów, współpracujący sprzęt, maszyny baz danych, pakiety użytkowe, tworzenie centrów danych. Obecnie uwagę swoją koncentrują na rozwoju kompletnego zestawu produktów do zarządzania danymi, w którym SZBD jest tylko jedną z części.

Pojawiają się na rynku pierwsze relacyjne SZBD (dotychczas systemy relacyjne miały charakter eksperymentalny i nie były dostępne na rynku), takie jak: ORACLE, ENCOMOPASS, INGRES. Dostawcy SZBD postanawiają włączyć do zastosowań swoich produktów języki relacyjne.

W ślad za rozwojem możliwości rozłożonego przetwarzania danych i pojawieniem się sprzętu wspomagającego, ponownie wzrasta zainteresowanie koncepcją SZBD wspierających takie przetwarzanie.

Kanadyjska filia firmy IBM występuje z obiecującą koncepcją zwaną CENTRUM Informacji (ang. Information Center). Nie prowadzone są natomiast prace standaryzacyjne. Firma IBM nie oferuje relacyjnych SZBD, a także nie pojawiają się wyspecjalizowane komputery firmy IBM do obsługi baz danych.

1981

E. F. Codd zdobywa po raz drugi Nagrodę Turinga, przyznaną również przez ACM za osiągnięcia w dziedzinie baz danych. Jest to rok, w którym święcą triumfy SZBD oparte na modelu relacyjnym. Jak na ironię do tego czasu zarówno Codd, jak i jego rywale reprezentujący podejście sieciowe, skonstruowali modele, które wybiegają daleko w stosunku do pierwotnych założeń.

Firma BRITTON-LEE prezentuje IDM-500, pierwszą opartą na modelu relacyjnym wyspecjalizowaną maszynę baz danych dla małych i średnich konfiguracji sprzętowych. Mimo stosunkowo wąskiego zakresu specjalizacji spotyka się ona z dużym zainteresowaniem.

Innowacje prezentowane przez Japończyków przyjmowane są z dużą uwagą.

1982

Firma IBM wypuszcza na rynek SZBD pod nazwą SQL/DS (ang. Structured Query Language Data System), przeznaczony do pracy pod nadzorem systemu operacyjnego DOS. Jest to przeznaczona na sprzedaż wersja SYSTEMU R, dysponująca językiem SEQUEL. Kwestią nierozstrzygniętą pozostaje termin, w którym firma IBM wprowadzi na rynek pakiet oprogramowania oparty na modelu relacyjnym, działający pod nadzorem systemu operacyjnego OS.

Model relacyjny wydaje się być preferowany przez dostawców maszyn baz danych, szczególnie przez firmę IBM. Firmy reprezentujące podejście relacyjne, takie jak: BRITTON-LEE, RELATIONAL SOFTWARE INC. — RSI (ORACLE), RELATIONAL TECHNOLOGY INC. — RTI (INGRES) obserwują bacznie pozycję rynkową firmy IBM.

Firma INTEL prezentuje na rynku relacyjną, wyspecjalizowaną maszynę baz danych — iDBP, przeznaczoną do automatyzacji prac biurowych. Chociaż zainteresowanie przetwarzaniem rozłożonym maleje w miarę pojawiania się ograniczeń związanych z obecną technologią SZBD, nie ustaje się w prowadzeniu prac w tej dziedzinie.

Zaczyna się mówić o pojawieniu się czwartej generacji języków. Z ogromnym zainteresowaniem spotykają się języki bezpośredniego użytkownika, takie jak: RAMIS, FOCUS, NOMAD, IDEAL oraz generatory programów użytkowych — MANTIS, ADS, ONLINE, DMS, ADF, UFO.

Koncepcja Centrów informacji staje się ważnym trendem w przemyśle.

Firma CULLINANE i inne robią pierwsze próby w kierunku połączenia minikomputerów i urządzeń graficznych z SZBD komputerów dużych.

Przyszłość

Lata osiemdziesiąte stanowią dziesięciolecie, w którym bazy danych staną się dominującą technologią zarządzania danymi. Chociaż do 1978 roku mniej niż 20% wszystkich systemów opierało się na tej technologii, przewiduje się, że w 1988 roku wskaźnik ten wyniesie 80%.

Oczekiwane jest pojawienie się nowych koncepcji w dziedzinie baz danych. Przyszłość technologiczna będzie związana z rozłożonym przetwarzaniem danych, SZBD wysokiego poziomu oraz odpowiednim sprzętem wspomagającym.

Opracowała GRAŻYNA SIARA
na podstawie DATA MANAGEMENT
nr 2, luty 1983

GIEŁDA INFORMACJI

NEPTUN 156

Gdańskie Zakłady Elektroniczne UNIMOR oferują monitory ekranowe NEPTUN 156, umożliwiające zobrazowanie znaków graficznych i alfanumerycznych z dużą rozdzielczością i jednoczesne odtwarzanie dźwięku towarzyszącego obrazowi. Przekątna ekranu z zieloną poświatą wynosi 12" (31 cm).

Monitor ten jest konstrukcją modułową, zapewniającą wygodę i łatwość serwisu. Na płycie czołowej umieszczony jest diodowy wskaźnik włączenia zasilania. Organy regulacji zewnętrznej (jaskrawość, kontrast, siła głosu) wykonane są na potencjometrach obrotowych. W ściance tylnej umieszczone jest wspólne gniazdo sygnałów wejściowych (gniazdo typu magnetofonowego).

Szczegółowych informacji udziela: mgr inż. Andrzej Sikorski, Dział handlowy GZE UNIMOR, ul. Rzeźnicka 54/56, 80-822 Gdańsk, tel. 31-03-71.

EST — ewidencja środków trwałych

Oferowany przez Zakłady Wytwórcze Przyrządów Pomiarowych i Systemów Minikomputerowych system ✱EST✱, zrealizowany na minikomputerze MERA-400, obejmuje całość zagadnień związanych z ewidencją środków trwałych w przedsiębiorstwie. Do głównych zadań należy: bieżąca aktualizacja kartotek na podstawie dokumentów dotyczących obrotu środkami trwałymi, ewidencjonowanie wszystkich zapisów księgowych związanych z gospodarką środkami trwałymi, automatyczne obliczanie, ewidencja i rozliczanie miesięcznych odpisów amortyzacji oraz umorzeń, automatyczna dekretacja amortyzacji i umorzeń, rozliczanie inwentaryzacji środków trwałych oraz szczegółowa kontrola formalna i merytoryczna dokumentów wraz z kontrolą dekretacji.

System ✱EST✱ dokonuje emisji szerokiej gamy wydawnictw: dla celów ewidencji księgowej, analizy działalności ekonomicznej przedsiębiorstwa, rozliczania spisu inwentaryzacyjnego, informacji kierownictwa oraz dla celów statystycznych i sprawozdawczości. Możliwe jest również szybkie wyszukiwanie informacji indywidualnych, np. o określonym obiekcie czy określonych zapisach księgowych. ✱EST✱ uwzględnia obowiązujące na dzień 1 stycznia 1983 r. przepisy z zakresu środków trwałych, klasyfikacji kosztów inwestycji, umorzeń i rozliczania amortyzacji.

Współpraca operatora z systemem odbywa się w prosty sposób konwersacyjny — tak, że osobą obsługującą system może być np. pracownik działu księgowości majątkowej. Dla średniej wielkości przedsiębiorstwa dane systemu ✱EST✱ mieszczą się na jednej kasecie wymiennej, co pozwala eksploatować system na standardowej minimalnej konfiguracji MERA-400.

Bliższych informacji udzielają: mgr mgr Jerzy T. Majewski i Krzysztof Wagner; ZWPPiSM, Warszawa, ul. Łopuszańska 117/123, tel. 23-76-11 wew. 593.

Zniszczyć CYFRONET?

Dokładnie dziesięć lat temu INFORMATYKA obwieściła[1]) uruchomienie Systemu Abonenckiego CYFRONET w Instytucie Badań Jądrowych w Świerku pod Warszawą. W czerwcu 1973 uruchomiono tam komputer CYBER 72 firmy CDC, który niebawem stał się głównym narzędziem informatycznym środowiska naukowego stolicy. Początkowo zainstalowano dziesięć urządzeń końcowych w uczelniach i instytutach naukowo-badawczych. Potem sukcesywnie wzrastały i zasięg, i jakość dokonywanych tu obliczeń numerycznych i przetwarzania danych.

Rok 1983, rok dziesięciolecia, stał się dla Środowiskowego Centrum Obliczeniowego rokiem klęski, niezrozumiałej o tyle, że reforma gospodarcza powinna mu, jako ośrodkowi dochodowemu i z inicjatywą, raczej sprzyjać. Nie dano mu jednak szans na rozwój. Państwowa Agencja Atomistyki (PAA) podjęła kilka decyzji prowadzących CYFRONET — jak się wydaje — do nikąd. Opiszmy jednak sprawy po kolei.

CYFRONET, będący do dzisiaj głównym ośrodkiem obliczeniowym naukowców Warszawy, opiera się na wysłużonym, dziesięcioletnim CYBERZE 73 (dzięki negocjacjom z CDC maszyna została rozbudowana do tego poziomu). Lata maszyny są oczywiście policzone. Zdobywanie części zamiennych wymaga dewiz. Dodatkowo sytuację utrudniają sankcje nałożone na PRL. Nie wystarcza tu wysiłek produkcji części i podzespołów własnym sposobem oraz adaptowanie i przyłączanie urządzeń produkcji krajowej (np. drukarka produkcji MERA-Błonie).

Podstawą dalszego bytu CYFRONETU postanowiono więc uczynić maszynę Jednolitego Systemu — RIAD 60 lub inną o zbliżonej mocy. Miała być ona zainstalowana w nowym budynku, który — budowany od 1974 roku, jako drugi etap rozwoju Centrum — zamierzano niebawem oddać do użytku; przynajmniej jego „komputerową" część (zamierzenia budowlane realistycznie ograniczono). W maju 1982 takie właśnie decyzje podjęła dyrekcja IBJ, przyjmując program działania CYFRONETU opracowany w dokumencie pt. „Problematyka komputerowa IBJ w latach 1982—1985".

Uznano, że budynek zostanie wykończony ze środków własnych. Na zakup nowego komputera i całkowite wykończenie budynku miał być zaciągnięty kredyt. Program ten został zaakceptowany w lipcu 1982 przez Zespół nadzorujący działalność bieżącą i rozwojową CYFRONETU, złożony z Podsekretarza Stanu w MNSzWiT, Prezesa PAA i Z-cy Sekretarza Naukowego PAN, a działający na podstawie „Porozumienia" z września 1977. Odpowiednie rozmowy w Banku Narodowym (oddział w Otwocku) oraz w Komisji Planowania zostały podjęte i zapowiadały powodzenie planu.

W grudniu 1982 przekształcono Instytut Badań Jądrowych (rozporządzeniem premiera) w trzy nowe instytuty atomistyki. Sytuacja CYFRONETU stała się wówczas niejasna. Można było, korzystając z okazji, doprowadzić do jego usamodzielnienia. Podporządkowanie go bezpośrednio PAA uregulowałoby bowiem problemy prawne, wynikające z „Porozumienia", za sprawą którego działalność CYFRONETU nadzorowana była przez Zespół na szczeblu Podsekretarzy Stanu. Podsekretarze ci weszli mocą ustawy sejmowej powołującej PAA w skład Zarządu Agencji. Usamodzielnienie byłoby — poza wszystkim — realizacją ogłoszonej reformy gospodarczej.

Dyrektor Centrum, prof. Roman Żelazny, składał kolejne propozycje: nadania CYFRONETOWI statusu ośrodka badawczo-rozwojowego atomistyki, potem zaś — przedsiębiorstwa państwowego usług informatycznych. Szczególnie druga propozycja wydawała się być rozwiązaniem korzystnym i — na czasie. Projekt ten został jednomyślnie poparty przez załogę Centrum, dającą w ten sposób votum zaufania swojemu dyrektorowi. Praktycznie bez uzasadnień obie propozycje odrzucono.

Wiosną bieżącego roku sytuacja stała się jeszcze bardziej niejasna. Decyzją Przewodniczącego Komisji do przeprowadzenia inwentaryzacji i podziału mienia b. IBJ z maja 1983, CYFRONETOWI odebrano budynek. Odwołanie do Prezesa PAA nie dało rezultatu. Nie nastąpiło spotkanie Zespołu nadzorującego ani Zarządu PAA, aby przedyskutować nową sytuację, mimo próśb i wniosków w tej sprawie. Podważono więc tym samym

[1]) Roman Żelazny: CYFRONET w Instytucie Badań Jądrowych w Świerku. Informatyka, nr 10/73, ss. 6—9

środowiskowy (a zatem ponadinstytutowy!) charakter Centrum. Prezes PAA zadecydował, że CYFRONET wejdzie w skład jednego z instytutów atomistyki — jako ... zakład doświadczalny.

Dyr. Żelaznemu polecono opracowanie projektu w takich ramach. Jednocześnie zadecydowano, że z Centrum zostanie wyłączony Zakład Obliczeń Reaktorowych, z którego praktycznie wywodzi się CYFRONET.

ZOR zajmował się teorią transportu neutronów i numerycznymi zagadnieniami obliczeń reaktorowych. W tej dziedzinie prof. Roman Żelazny jest autorytetem w skali międzynarodowej; powszechnie uważany jest za twórcę polskiej szkoły teorii transportu neutronów i obliczeń reaktorowych. Pod względem naukowym ZOR stanowił integralną część Centrum, nadającą mu ważną dla atomistyki specjalizację (równania różniczkowe cząstkowe oraz całkowo-różniczkowe).

Trudno się więc dziwić, że — nie otrzymawszy żadnych wskazówek co do dalszych losów całego CYFRONETU — dyr. Żelazny uznał, że wykonanie takiego polecenia jest niemożliwe. W lipcu br. został odwołany ze stanowiska.

Pozbawiony swego twórcy i animatora, mającego autorytet naukowy i potrafiącego skutecznie gospodarzyć nawet w tak trudnych jak obecnie, embargowych warunkach, bez szans na nowy komputer i pomieszczenie dla niego, osłabiony intelektualnie odejściem kadry ZOR, traktowany jak zawadzający podrzutek, CYFRONET stanie się zapewne informatycznym memento reformy gospodarczej. Jej zbyt już wyraźnym zaprzeczeniem.

CYFRONET jest środowisku naukowemu potrzebny. Świadczą o tym choćby niedawne poparcia składane przez Polską Akademię Nauk, czy wyższe uczelnie Warszawy. Dyskutowane były projekty rozszerzenia zadań CYFRONETU dla zaspokojenia potrzeb obliczeniowych Instytutu Meteorologii i Gospodarki Wodnej. Instytut ten musi bowiem rozbudować swoje moce obliczeniowe kosztem znacznych nakładów. Wykorzystanie budynku CYFRONETU i jego zasobów komputerowych dałoby tu znaczne oszczędności, zaś samemu Centrum — potężne wsparcie finansowe. W tej sprawie PAA nie podjęła nawet dyskusji.

Jeden z liczących się ośrodków informatycznych w Polsce skazano na upadek. Trudno zrozumieć motywy osób, które podjęły decyzję, a które — nawiasem mówiąc — chętnie powołują się na interes społeczny. Nie widać dzisiaj jakiejś wyraźniejszej koncepcji zaspokojenia potrzeb informatycznych atomistyki i środowiska naukowego stolicy; sami zainteresowani także niewiele potrafią powiedzieć.

Kto wie, może „atomistyka na liczydłach" stanowi jeden z obowiązujących elementów zreformowanego kraju.

ZBIGNIEW GLUZA

mikroKLAN

Mikroprocesory przestały być w Polsce elitarnym hobby. Pierwsze „prawdziwe" systemy mikroprocesorowe stały się rzeczywistością. Ale są jeszcze absurdalnie drogie. Zmusza to projektantów sprzętu i oprogramowania do stosowania niekonwencjonalnych rozwiązań, własnych lub przejętych od kogoś — stwarzających rynek nie całkiem zależny od państwowego producenta...

mikroKLAN

— od nowego roku wkładka INFORMATYKI, poświęcona popularyzacji „mikroprocesorowych" pomysłów. Znajdą w niej miejsce ciekawe rozwiązania sprzętowe i programowe. A także opisy „ukrytych" cech popularnych układów, pomijanych w katalogach, a przysparzających konstruktorom wielu kłopotów. Na łamy trafią wszelkie użyteczne propozycje profesjonalistów i hobbystów.

Zapraszamy do mikroKLANU.

GIEŁDA INFORMACJI

EDYCJA EKRANOWA MOP

Centralny Ośrodek Informatyki Górnictwa oferuje oprogramowanie umożliwiające w systemie MOP edycję ekranową (komenda SD) przy użyciu lokalnych monitorów ekranowych MERA 7911, dołączonych za pośrednictwem sterownika MERA — JSG 7802 do komputera ODRA pracującego pod kontrolą systemu operacyjnego GEORGE-3 (wersja 865 i wyższe). Brak pełnej zgodności między sterownikiem MERA a jego odpowiednikiem angielskim CCU 7180, uniemożliwia stosowanie oryginalnego oprogramowania firmy ICL (ELWRO) przeznaczonego do realizacji edycji ekranowej.

Oferowane oprogramowanie dostosowuje format komunikatów wyprowadzanych przez system GEORGE-3 do postaci akceptowalnej przez system monitorów lokalnych MERA. Składa się ono z dwóch programów:

— zmodyfikowanej wersji „Menedżera komunikacyjnego" wygenerowanego dla danej konfiguracji monitorów ekranowych

— programu, o pojemności 1 K słów pamięci operacyjnej, rezydującego na stałe w tej pamięci.

Bliższych informacji udziela mgr inż. Marek Kwiatkowski; Zakład BS, COIG, Katowice, ul. Mikołowska 100, tel. 57-41-50.

Klub mikrokomputerowy ABAKUS

Klub został założony w Warszawie przez Leszka Wilka w maju 1983 jako Sekcja Klubu Osiedlowego Wilanów. Sprzęt, w postaci trzech telewizorów oraz czterech mikrokomputerów: SPECTRUM (48 K), dwóch ORIC-1 oraz ZX-81 (16 K) został Klubowi wypożyczony.

Członkiem Klubu, zgodnie z tymczasowym regulaminem, może zostać każdy, kto wyrazi ku temu chęć oraz przedstawi Radzie Klubu projekt programu na odpowiednim poziomie. Członkostwo uprawnia do swobodnego korzystania ze sprzętu, czasopism oraz literatury, dostępnych w Klubie.

ABAKUS nawiązał kontakty z trzema firmami polonijnymi: IMPOL II w Warszawie, MARCO-ELECTRONIC w Krakowie i AMEPROD w Poznaniu oraz z firmą rzemieślniczą „Computer Studio Kajkowski" w Gdyni, otrzymując poparcie i obietnicę pomocy materialnej.

W drugiej połowie sierpnia Klub ogłosił otwarte konkursy na pomysł i program gry edukacyjnej z nagrodami w postaci dwóch mikrokomputerów ZX-81 (p. INFORMATYKA nr 9/83), zorganizował też dwa turnieje szachowe (w Warszawie i Krakowie) z udziałem mikrokomputerów.

W każdy poniedziałek od 17 do 20 Klub jest otwarty dla szerokiej publiczności.

Adres: Klub Mikrokomputerowy ABAKUS; 02-930 Warszawa, ul. Sobieskiego 18, tel. 42-91-85.

Seminarium minikomputerowe

Czechosłowacki sprzęt minikomputerowy, zainstalowany w Politechnice Warszawskiej, stał się podstawą wiosennego (12—14 kwietnia 1983) seminarium naukowo-technicznego „Minikomputery SM 4-20 oraz ich zastosowanie". Cel spotkania był jednak o wiele szerszy — miało ono wyczerpująco zaprezentować mini- i mikrokommutery należące do rodziny SMEP (System Malych Elektronickych Počitačov).

Seminarium zostało zorganizowane przez czechosłowackie przedsiębiorstwo handlu zagranicznego PZO KOVO z Pragi przy współudziale warszawskiego Instytutu Techniki Lotniczej. Prezentacji dokonali przedstawiciele firmy DATASYSTEM z Bratysławy, przy czynnym uczestnictwie polskich użytkowników tych maszyn. Zainteresowanie tematyką seminarium przerosło najśmielsze oczekiwania organizatorów; pierwszego dnia zamiast spodziewanych kilkudziesięciu uczestników, na sali pojawiło się ok. 200 osób.

Tematykę seminarium podzielono na dwie części. Do pierwszej można zaliczyć ogólny opis sprzętu oraz systemów operacyjnych, do drugiej natomiast — przykłady zastosowań minikomputerów w różnych dziedzinach (w projektowaniu, służbie zdrowia, symulacji i grafice komputerowej oraz zbieraniu danych). W trakcie seminarium odbyły się prezentacje niektórych spośród omawianych rozwiązań. Było to możliwe dzięki minikomputerowi SM 4-20, zainstalowanemu w Instytucie Techniki Lotniczej PW — będącym, jak można wnioskować z bezpośrednich rozmów, bardzo cenionym narzędziem w codziennych pracach Instytutu.

Ogólny opis urządzeń

Rodzina mini- i mikrokomputerów dzieli się na trzy gałęzie: SMEP I, SMEP II, SMEP III.

Najwcześniejszym produktem są minikomputery należące do pierwszej grupy. Na seminarium omówiono dwa z jej przedstawicieli: SM 3-20 i SM 4-20. Producenci nie przewidują dalszego rozwoju tej grupy.

Mini- i mikrokomputery SMEP II stanowią grupę nowszych urządzeń, charakteryzujących się m.in. lepszymi rozwiązaniami technicznymi i diagnostyką oraz bogatszym zestawem rozkazów. Jest ona także znacznie bardziej rozbudowana od pierwszej grupy, przy czym zachowano kompatybilność w górę. Grupa SMEP II składa się z pięciu podgrup, z numeracją SM 50 do SM 54, a każda z nich dzieli się jeszcze na mniejsze. I tak — na przykład — w ramach podgrupy SM 50 opartej na mikroprocesorze INTEL 8080 można wyróżnić sześć modeli z których dwa (SM 50/40 i SM 50/50) były omawiane podczas seminarium. W grupie SMEP II obok rozwiązań już funkcjonujących i dopracowanych znajdują się modele będące przedmiotem badań.

Ostatnia z wymienionych grup — SMEP III istnieje dopiero w postaci planów. Ze wstępnych doniesień wynika, że będą to urządzenia bardziej specjalizowane, oparte w całości na technice mikroprocesorowej. Ze względu na brak konkretnych informacji na temat SMEP III, grupa ta na seminarium nie była w ogóle omawiana.

Wszystkie prezentowane komputery mogą być przyłączane do wspólnej magistrali UNIBUS. Przy takim rozwiązaniu pamięć operacyjna traktowana jest — jak wiadomo — na równi z innymi urządzeniami zewnętrznymi. Daje to wiele korzyści, jak choćby — możliwość rozbudowywania pamięci bez zmian organizacji procesora, możliwość dołączania modułów pamięci o różnych szybkościach czy szansę zwiększenia standaryzacji konstrukcji jednostek sterujących. Pojemność pamięci operacyjnych jest dla poszczególnych modeli różna i waha się od 32 K słów 16-bitowych (SM 3-20), przez 128 K słów (SM 4-20 i SM 52/11), aż po 4 M bajty (SM 52/11 PLUS — potencjalnie możliwe dzięki 22-bitowej szynie adresowej). Czas dostępu jest rzędu 0,5 mikrosekundy.

Dodatkowym udogodnieniem jest jednostka zarządzania pamięcią (MMU — Memory Management Unit), pozwalająca w części prezentowanych modeli wykorzystać pamięć wirtualną oraz dostępną dla modeli SMEP II bardzo szybką pamięć CACHE o pojemności 1 K słów. Według opinii jednego z wykładowców, kompilacja przy zastosowaniu tego modułu jest siedmiokrotnie szybsza niż w przypadku stosowania dotychczasowych rozwiązań.

Producenci sprzętu gwarantują możliwość dołączenia następujących urządzeń zewnętrznych:

— pamięci dyskowej kasetowej CM 5400 o pojemności 2,4 M bajtów (poprzez jednostkę sterującą CM 5105)
— pamięci dyskowej CM 5405 o pojemności 29 MB
— pamięci taśmowej CM 5300 (poprzez jednostkę sterującą CM 5001)
— pamięci na dyskach elastycznych CM 5605 o pojemności 512 K bajtów
— adapterów PAD-8, PAD-12 i PAD-16
— perforatora taśmy CM 6208
— drukarki mozaikowej z klawiaturą CM 6303
— drukarki wierszowej CM 6313 o prędkości 600 wierszy/min
— adaptera asynchronicznego CM 6002 do obsługi jednego terminala z prędkością do 9600 b/s
— adaptera asynchronicznego CM 8606 do połączeń minikomputer—minikomputer i minikomputer—komputer
— czytnika kart dziurkowanych EC 6112 o prędkości 300 kart/min
— terminala numerycznego CM 9402
— alfanumerycznego monitora CM 7202 (84 × 24) przeznaczonego dla systemów konwersacyjnych
— sterownika CM 9004 wraz z czujnikami CM 9104 do zbierania i wysyłania sygnałów analogowych i cyfrowych
— urządzeń rysujących CM 1712 i CM 1208.

Dowolny model rodziny SMEP I lub SMEP II umożliwia więc elastyczne tworzenie konfiguracji zgodnych z konkretnymi potrzebami użytkownika.

Systemy operacyjne

Sporo uwagi poświęcono systemom operacyjnym przeznaczonym dla maszyn SMEP I i SMEP II.

Dla pierwszej przewidziany jest m.in. system FOBOS-1. Umożliwia on w zasadzie pracę jednoprogramową z możliwością pracy w czasie rzeczywistym oraz wsadowego przetwarzania danych. System operacyjny FOBOS-1 zawiera kompilatory języków BASIC, FORTRAN i MAKROASSEMBLER.

Druga wersja tego systemu — FOBOS-2 zawiera, obok wymienionych, translatory języków BASIC-S, FOKAL oraz VU BASIC. Ten ostatni stwarza możliwość równoległej pracy w czasie rzeczywistym ośmiu użytkownikom. W przeciwieństwie do pierwszej wersji systemu jest on przeznaczony dla maszyn z pamięcią o pojemności ponad 32 K słów.

System operacyjny DIAMS jest wielozadaniowym systemem konwersacyjnym. Przewiduje się jego zastosowanie do celów:

- operatywnego zarządzania
- kierowania eksperymentami naukowymi
- automatyzacji prac projektowych
- przetwarzania danych na potrzeby zarządzania
- zbierania, ochrony i przetwarzania informacji z zastosowaniem technologii bazy danych, wykorzystywanej w trybie wielodostępu z lokalnych i zdalnych terminali.

System ten stwarza — między innymi — możliwość pracy wieloprogramowej, tworzenie bazy danych o strukturze hierarchicznej, o pojemności informacyjnej do 200 M bajtów, jednoczesny dostęp do bazy z maksymalnie 48 terminali, autoryzację dostępu do programów i danych, organizację rozproszonych baz danych w systemach wielomaszynowych.

Podobnie jak FOBOS, system ten istnieje w dwóch wersjach — DIAMS-1 i DIAMS-2. Pierwsza z nich jest przeznaczona dla wcześniejszych modeli maszyn, druga zaś dla maszyn nowszych. Obie wersje nie są jednak kompatybilne. Systemy zawierają m.in. translatory języków MUMPS (różnych dla każdej z wersji) oraz programy obsługi urządzeń wejścia i wyjścia.

Kolejny jest system operacyjny DOS RVR. Jest to produkt umożliwiający jednoczesną pracę wielu użytkowników. Ma on szeroki zestaw kompilatorów języków: MAKROASSEMBLER, FORTRAN, FORTRAN IV PLUS, BASIC (interpreter), BASIC-PLUS, COBOL, BPS oraz PASCAL (w przygotowaniu). Autorzy tego systemu przewidują jego zastosowanie w automatyzacji prac naukowo-badawczych i projektowo-konstrukcyjnych, zbieraniu i analizie informacji — z możliwością pracy wieloprogramowej w czasie rzeczywistym.

Producenci oferują także system DOS MVS przeznaczony dla mikrokomputerów SM 50/50 i SM 50/40 z kompilatorami języków PL/M-80, FORTRAN 80, DOS-MAKROASSEMBLER i PROCES BASIC-80. System ten umożliwia użycie tych mikrokomputerów do sterowania niewielkich procesów technologicznych oraz jako elementów sieci komputerowych i terminalowych stacji zbierania danych.

Przykłady zastosowań

Podczas seminarium przedstawiono także kilka przykładów zastosowania rozwiązań sprzętowych i programowych. Jednym z nich jest wykorzystanie w szpitalu dwóch minikomputerów SM 4-20 oraz mikrokomputera SM 50/50. System ten obejmuje m.in.:

- przetwarzanie danych dotyczących pacjenta, powstałych przy: jego przyjmowaniu i zwalnianiu, podczas leczenia, wydawaniu diagnoz lekarskich, w trakcie tworzenia statystyk; ta część systemu zawiera także przetwarzanie danych na potrzeby bieżącego zarządzania szpitalem
- zbieranie i analiza informacji z badań elektrokardiograficznych i spirometrycznych
- zbieranie danych z laboratoriów, powiązane z aktualizacją bazy danych.

Rozwiązania zaprezentowane w dwóch ostanich punktach zakładają przyłączenie sprzętu komputerowego (poprzez czujniki) do aparatury medycznej.

Przykładem narzędzia ułatwiającego zastosowanie minikomputerów rodziny SMEP jest także język LAHYSS, przedstawiony przez przedstawiciela Instytutu Podstawowych Problemów Techniki PAN. Jest to język służący do modelowania układów dynamicznych, opisanych równaniami różniczkowymi lub schematem blokowym. Język ten jest ukierunkowany na potrzeby projektowania inżynierskiego, badań podstawowych oraz dydaktyki. LAHYSS zapewnia skrócenie czasu obliczeń w stosunku do dotychczasowych metod, łatwość programowania, rozbudowaną diagnostykę błędów oraz konwersacyjny tryb pracy.

Kolejny przykład zastosowania minikomputerów to GOLEM — system zbierania danych z monitorów ekranowych (maksymalnie 16). Rozwiązanie to jest zbliżone do produkowanego przez nasz przemysł systemu MERA 9150.

* * *

Przedstawione przykłady nie wyczerpują — ze zrozumiałych względów — pełnego zestawu zaprezentowanych rozwiązań, ani tym bardziej — pełnych możliwości wykorzystania omawianego sprzętu. Tak zestawy mikrokomputerowe rodziny SMEP (tworzone z reguły dla konkretnych zastosowań), jak i minikomputerowe (z reguły bardziej uniwersalne) stwarzają o wiele szerszą gamę udogodnień.

Spotkania tego typu są bardzo potrzebne. W przyszłości wskazane byłoby zapewnienie tłumaczenia — przynajmniej części wykładu. Podczas tegorocznego seminarium ta niedogodność, pomimo dużych podobieństw języka słowackiego i polskiego, w połączeniu z koniecznością posługiwania się terminologią techniczną oraz niezbyt bogatym zestawem materiałów informatycznych — w dużym stopniu utrudnia odbiór przedstawionych treści.

Wspomniane uchybienia — zrozumiałe wobec zaskoczenia organizatorów dużym zainteresowaniem tematyką seminarium — nie przekreśliły wszakże ani jego rangi, ani sensu.

ANNA i WITOLD LUCIŃSCY

Konkursy oszczędności

Urząd Gospodarki Materiałowej wspólnie z Ministerstwem Nauki, Szkolnictwa Wyższego i Techniki, Naczelną Organizacją Techniczną oraz Polskim Towarzystwem Ekonomicznym przystępują do organizowania **KONKURSÓW OSZCZĘDNOŚCI SUROWCÓW I MATERIAŁÓW** dla przedsiębiorstw, jednostek zaplecza naukowo-badawczego oraz wynalazców i racjonalizatorów.

Konkursy będą miały charakter powszechny, a prace konkursowe zgłaszane mają być w dwóch kategoriach:

- zakładów uspołecznionych
- osób indywidualnych lub zespołów tych osób.

Do konkursów będzie można zgłaszać przedsięwzięcia podjęte po 1 stycznia 1983 r. Przedsięwzięcia muszą być wdrożone najpóźniej w okresie dwóch lat od ogłoszenia danego konkursu. Decydującym kryterium oceny zgłoszonych prac będzie efektywność ekonomiczna wdrożonego przedsięwzięcia. Zgłoszone prace oceniać będą sądy konkursowe powołane przez Ministra Gospodarki Materiałowej w porozumieniu z Ministrem Nauki, Szkolnictwa Wyższego i Techniki. Sekretariat konkursów prowadzić będzie Instytut Gospodarki Materiałowej, Al. Jerozolimskie 30, 00-024 Warszawa. Przy sekretariacie konkursów zostaną powołane specjalne komisje kwalifikacyjne, które dokonają weryfikacji zgłoszonych rozwiązań zarówno pod względem merytorycznym, jak i zgodności z warunkami konkursów i wraz z opinią przedstawią je właściwym sądom konkursowym.

Za najlepsze wdrożone prace, ocenione przez sądy konkursowe, przyznane zostaną w obu wyżej wymienionych kategoriach nagrody I, II, III i IV stopnia. Ponadto przyznane zostaną nagrody specjalne i wyróżnienia. Nagrody będą wypłacane wyłącznie po wdrożeniu przedsięwzięcia i uzyskaniu efektów ekonomicznych. Szczegółowe warunki konkursów zostaną opublikowane w najbliższym czasie.

Historia warszawskiego ETOBU (1953–1983)

Historia Warszawskiego Przedsiębiorstwa Informatyki Przemysłu Budowlanego ETOB wiąże się ściśle z powstaniem krajowej sieci ETOB i rozwojem zastosowań techniki obliczeniowej w polskim budownictwie. Jest to jakby przedsiębiorstwo — matka, które utworzyło całą sieć jednostek ETOB, a także tworzyło jednolite systemy informatyczne, koordynowało ich zastosowanie i rozwój w budownictwie oraz przemyśle materiałów budowlanych na terenie całego kraju. Dopiero w wyniku dalszego rozwoju terenowych jednostek ETOB nastąpiło ich pełne usamodzielnienie. Na przestrzeni wielu lat doprowadziło to do ukształtowania się sieci przedsiębiorstw w miastach stanowiących ważniejsze zagłębia budowlane.

W roku bieżącym minęło 30 lat od czasu, kiedy ówczesne Ministerstwo Budownictwa Przemysłowego podjęło decyzję o utworzeniu jednostki organizacyjnej zajmującej się zastosowaniem techniki obliczeniowej w pracach administracyjno-biurowych i w zarządzaniu jednostkami gospodarczymi resortu budownictwa. Była to wówczas bardzo śmiała decyzja, albowiem technika obliczeniowa była mało rozpowszechniona i stosowana na szerszą skalę tylko w kilku jednostkach w kraju. W budownictwie technika ta w ogóle nie była wówczas znana, a przyszli jej użytkownicy (przedsiębiorstwa i zjednoczenia budownictwa) w większości odnosili się do niej z rezerwą.

Nie dysponując ani odpowiednim lokalem, ani właściwym sprzętem informatycznym przystąpiono do tworzenia planów informatycznej obsługi przedsiębiorstw, zjednoczeń i centrali Ministerstwa. Na mocy zarządzenia ministra Budownictwa Przemysłowego z 30 lipca 1952 r. utworzono przedsiębiorstwo państwowe pod nazwą Biuro Rozliczeń Budownictwa Przemysłowego z siedzibą w Warszawie. Przedmiotem działania Biura miało być — zgodnie z zarządzeniem — „prowadzenie ewidencji i gospodarki materiałowej, rozliczeń robocizny i kosztów własnych za pomocą maszyn statystyczno-rachunkowych, opracowywanie projektów organizowania rozliczeń oraz instruktaż w zakresie koniecznej dokumentacji w odniesieniu do jednostek organizacyjnych podległych Ministrowi Budownictwa Przemysłowego".

Organizowanie przedsiębiorstwa rozpoczęto w styczniu 1953. Jego siedzibą był budynek przy ul. Józefa Bema 65 w Warszawie, w którym do dziś mieści się Warszawskie Przedsiębiorstwo Informatyki Przemysłu Budowlanego ETOB. Równolegle z organizacją Przedsiębiorstwa w Warszawie, utworzono oddział tego przedsiębiorstwa w Krakowie. Na skutek reorganizacji ówczesnych resortów budownictwa, od stycznia 1961 przedsiębiorstwo warszawskie zmieniło swoją nazwę na Biuro Rozliczeń Budownictwa ROZBUD.

Pierwsze maszyny zainstalowano w trzecim kwartale 1953 i na nich w ciągu czwartego kwartału tego roku wykonano próbne tabulogramy. Były to maszyny licząco-analityczne marki SAM pracujące w oparciu o karty 45-kolumnowe (jeden zestaw) i 80-kolumnowe (także jeden zestaw). W tym czasie analogiczny park maszynowy znajdował się w oddziale w Krakowie. W 1954 r. zainstalowano dalsze dwa zestawy maszyn w Warszawie (jeden 80-kolumnowy i jeden 45-kolumnowy) oraz takie same dwa zestawy maszyn w Oddziale w Krakowie.

W końcu 1955 podjęto decyzję ujednolicenia parku maszynowego — w Warszawie zgrupowano maszyny pracujące na kartach 80-kolumnowych, a w Oddziale w Krakowie — na 45-kolumnowych. W ten sposób ukierunkowano: przedsiębiorstwo w Warszawie na maszyny 80-kolumnowe marki SAM, a następnie marki BULL; Oddział w Krakowie — na maszyny 45-kolumnowe marki SAM i następnie 90-kolumnowe marki ARITMA, a w późniejszym okresie maszyny 80-kolumnowe marki SAM. W Warszawie, po próbnej eksploatacji, od stycznia 1954 rozpoczęto normalną obsługę trzech przedsiębiorstw budowlanych w zakresie systemu gospodarki materiałowej.

Stopniowo rozszerzano zastosowania techniki MLA na inne jednostki resortu. Do 1958 r. nie było jednak istotnego postępu zarówno w ilości eksploatowanego sprzętu, jak i w jego zastosowaniu. W dalszym ciągu wykonywano usługi jedynie w zakresie gospodarki materiałowej, przy czym już wówczas podjęto prace nad udoskonaleniem tego systemu. W wyniku tego powstał system gospodarki materiałowej oparty na jednolitym resortowym indeksie materiałowym, który z pewnymi zmianami (wynikającymi z aktualnych przepisów) i odpowiednim oprogramowaniem, ze względu na inny sprzęt, jest eksploatowany w całej sieci przedsiębiorstw ETOB oraz w wielu innych ośrodkach obliczeniowych w kraju.

Dopiero po 1958 r., kiedy zwiększono dostawy maszyn (ze Związku Radzieckiego, Czechosłowacji i Francji) nastąpił ilościowy rozwój sprzętu i jego zastosowań. W istotny sposób przyczyniły się do tego również nowe, znacznie sprawniejsze tzw. mnożarki, które wspierały zestawy MLA. W tym czasie wzrosło zapotrzebowanie na technikę obliczeniową i wynikła potrzeba tworzenia nowych oddziałów Biura Rozliczeń Budownictwa.

W 1960 r. BRB w Warszawie przystąpiło do organizacji oddziału w Poznaniu, aby uruchomić go w 1961 r. Po doprowadzeniu tego oddziału do pełnej sprawności, przystąpiono do organizacji ośrodka w Katowicach, który od stycznia 1964 do czerwca 1976 działał w ramach organizacji Śląskiego Zjednoczenia Budownictwa Przemysłowego, a od lipca 1976 organizacyjnie został podporządkowany Centrum Informatyki Przemysłu Budowlanego ETOB. W następnych latach przystąpiono do organizacji oddziałów w Łodzi i Bydgoszczy. Oddział w Łodzi rozpoczął swoją działalność w 1967, a ponadto Gdańskie Zjednoczenie Budownictwa powołało w 1969 r. własny ośrodek obliczeniowy, który w 1972 r. przejęło Centrum ETOB.

Jednocześnie z rozszerzeniem zastosowań techniki MLA, od 1961 r., zorganizowano obsługę przedsiębiorstw za pomocą średniej techniki obliczeniowej. Technika ta była i jest jeszcze w Polsce szeroko stosowana, szczególnie w przedsiębiorstwach przemysłu materiałów budowlanych.

1 stycznia 1969 Biuro Rozliczeń Budownictwa przekształciło się w Centrum Elektronicznej Techniki Obliczeniowej Przemysłu Budowlanego ETOB, a jego oddziały terenowe stały się Zakładami Obliczeniowymi. Nowa nazwa wiązała się z nowymi kierunkami rozwoju sprzętu informatycznego, a przede wszystkim z elektroniczną techniką obliczeniową. W tym czasie w sieci ETOB pracowało 70 zestawów MLA, wspieranych elektronicznymi maszynami typu GAMMA 3 i ODRA 1113. W Warszawie pracowało wówczas 14 zestawów wraz z odpowiednim sprzętem uzupełniającym. Na szeroką skalę podjęto wówczas prace przygotowawcze do instalacji i eksploatacji dużych maszyn cyfrowych.

1 stycznia 1970 do Centrum ETOB włączono ośrodek Zastosowań Elektronicznej Techniki Obliczeniowej ETOPROJEKT, powołany w listopadzie 1962. ETOPROJEKT działał przy Zjednoczeniu Biur Projektów Budownictwa w ramach branżowego zaplecza badawczo-rozwojowego. W końcu 1964 wyposażony został w EMC ZAM-2 GAMMA. Włączając ETOPROJEKT do Centrum ETOB stworzono jedną

wyspecjalizowaną jednostkę zajmującą się informatyką w przemyśle budowlanym.

W wyniku dalszej reorganizacji, 1 lipca 1972 dawne BRB, a późniejsze Centrum ETOB przekształciło się w Warszawskie Przedsiębiorstwo Informatyki Przemysłu Budowlanego ETOB, wszystkie zakłady obliczeniowe podniesiono do rangi przedsiębiorstw oraz utworzono Centrum Informatyki Przemysłu Budowlanego ETOB, pełniące funkcję zjednoczenia.

Do zadań nowo powstałego CIPB, poza problematyką zarządzania i obliczeniami inżynierskimi, włączono jeszcze sterowanie procesami produkcyjnymi. Dążąc do dalszego rozwoju zastosowań sprzętu informatycznego w budownictwie, w 1971 r. przy Centrum ETOB powołano Zakład, który w 1972 r. przekształcono w Pracownię Projektowania Systemów Informatyki i Doradztwa Organizacyjnego w Budownictwie ETOBSYSTEM. Pracownia ETOBSYSTEM działała przy Centrum ETOB do 31 lipca 1976 r. i nie posiadała własnego komputera. Od 1 sierpnia 1976 r. Pracownia ETOBSYSTEM uległa likwidacji i została przejęta przez przedsiębiorstwo warszawskie.

W 1976 r. powołano Rzeszowskie Przedsiębiorstwo Informatyki Przemysłu Budowlanego ETOB — na bazie zakładu obliczeniowego w Rzeszowie, zorganizowanego przy przedsiębiorstwie krakowskim.

W styczniu 1977 Warszawskie Przedsiębiorstwo Informatyki Przemysłu Budowlanego ETOB przejęło ośrodek obliczeniowy w Białymstoku od przedsiębiorstwa w Gdańsku. Ośrodek ten był wyposażony w dziurkarki i sprawdzarki kart marki SAM. W stosunkowo krótkim czasie rozwinął się i już w końcu 1977 wyposażony został przez ETOB w Warszawie w komputer ODRA 1305 i inny sprzęt towarzyszący, a w 1978 r. rozpoczął pełną obsługę informatyczną na własnym komputerze w oparciu o oprogramowanie i technologię przejętą z przedsiębiorstwa warszawskiego. W 1978 r. ośrodek obliczeniowy w Białymstoku przekształcono w zakład obliczeniowy, działający w ramach WPIPB ETOB, by w październiku 1982 przekształcić go w samodzielne przedsiębiorstwo pod nazwą Białostockie Przedsiębiorstwo Informatyki Przemysłu Budowlanego ETOB. Jako samodzielne przedsiębiorstwo działało ono do marca 1983, a od kwietnia 1983 weszło w skład Centrum Informatyki Przemysłu Budowlanego ETOB.

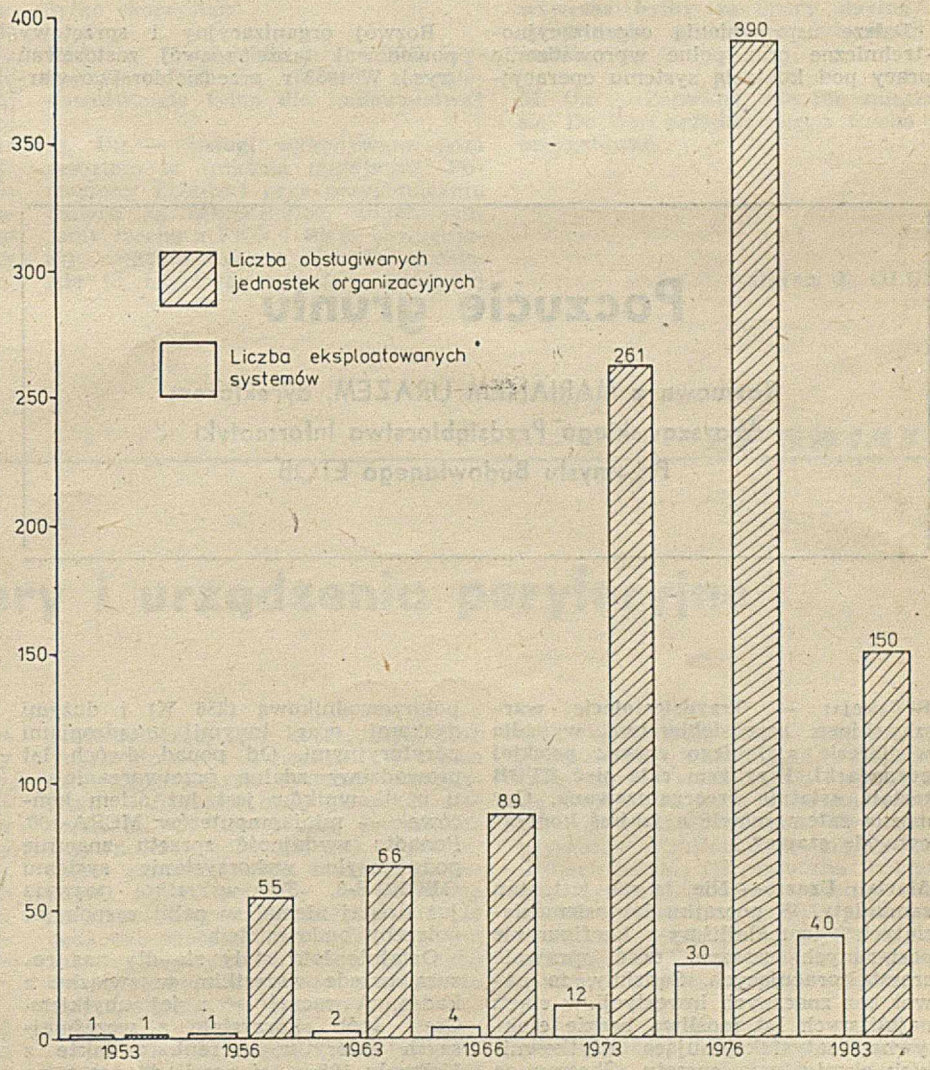
WPIPB ETOB zorganizowało także ośrodek obliczeniowy w Lublinie. Pierwsze prace podjęto w 1973 r., tworząc przy pomocy Lubelskiego Zjednoczenia Budownictwa stację przygotowania danych na terenie Lubelskiego Przedsiębiorstwa Budownictwa Miejskiego, w grudniu 1974 podjęto natomiast decyzję o powołaniu ośrodka obliczeniowego ETOB w Lublinie, przyporządkowanego przedsiębiorstwu w Warszawie. Ośrodek ten w końcu 1978 r. został wyposażony w komputer ODRA 1305 oraz sprzęt towarzyszący.

Ośrodek w Lublinie stosunkowo szybko rozwinął się i w styczniu 1980 został przekształcony w zakład obliczeniowy podległy w dalszym ciągu WPIPB ETOB. Taki stan organizacyjny trwał do lipca 1981, a następnie podporządkowano go Centrum Informatyki Przemysłu Budowlanego ETOB.

Centrum ETOB w wyniku wprowadzenia reformy gospodarczej utraciło funkcje zejdnoczenia i ukierunkowało swoją działalność przede wszystkim na obsługę przemysłu materiałów budowlanych. W ślad za podporządkowaniem sobie zakładu w Lublinie, Centrum ETOB przejęło także zakład w Olsztynie — podległy gdańskiemu przedsiębiorstwu, w Kielcach (podległy przedsiębiorstwu krakowskiemu), w Koszalinie (bydgoskiemu) oraz ośrodek w Opolu — z przemysłu cementowego.

Ukształtowało to obecnie istniejącą organizację sieci przedsiębiorstw ETOB. Aktualnie w jej skład wchodzi:

- dobrowolne Zrzeszenie Przedsiębiorstw Informatyki Przemysłu Budowlanego ETOB, do którego należą następujące przedsiębiorstwa:
  — Bydgoskie Przedsiębiorstwo Informatyki Przemysłu Budowlanego ETOB
  — Katowickie Przedsiębiorstwo Informatyki Przemysłu Budowlanego ETOB
  — Łódzkie Przedsiębiorstwo Informatyki Przemysłu Budowlanego ETOB
  — Poznańskie Przedsiębiorstwo Informatyki Przemysłu Budowlanego ETOB
  — Warszawskie Przedsiębiorstwo Informatyki Przemysłu Budowlanego ETOB
  — Wrocławskie Przedsiębiorstwo Informatyki Przemysłu Budowlanego ETOB
- Przedsiębiorstwa nie zrzeszone, do których należą:
  — Centrum Informatyki Przemysłu Budowlanego ETOB



Rozwój usług informatycznych WPIPB ETOB w latach 1955—1983

— Gdańskie Przedsiębiorstwo Informatyki Przemysłu Budowlanego ETOB — Krakowskie Przedsębiorstwo Informatyki Przemysłu Budowlanego ETOB — Rzeszowskie Przedsiębiorstwo Informatyki Przemysłu Budowlanego ETOB.

Rozwój przedsiębiorstwa warszawskiego opierał się na potencjale sprzętowym i kadrowym zgrupowanym w Warszawie. Maszyny licząco-analityczne z lat pięćdziesiątych przestały wystarczać. W 1970 r. zakupiono więc komputer MIŃSK 32, a w niespełna rok później komputer ODRA 1304. Zainstalowanie dwóch komputerów w Warszawie umożliwiło stopniowe wycofywanie zestawów MLA. W 1975 r. zakupiono jeszcze dwa komputery ODRA 1305. Umożliwiło to całkowite wycofanie techniki MLA; ostatni zestaw tych maszyn wycofany został w 1976 r.

Dalszy rozwój sprzętu następował przede wszystkim poprzez rozbudowę dwóch komputerów ODRA 1305, aż do maksymalnych konfiguracji. Umożliwiło to wycofanie w latach 1981—1982 z eksploatacji zużytych komputerów MIŃSK 32 i ODRA 1304.

Dalsze usprawnienia organizacyjno-techniczne oraz pełne wprowadzenie pracy pod kontrolą systemu operacyjnego GEORGE 3 — umożliwiły także rozwój teleprzetwarzania. Aktualnie przedsiębiorstwo posiada osiem końcówek u pięciu użytkowników. Jako końcówki podłączone są minikomputery MERA-100, które są wykorzystywane do zdalnego i lokalnego przetwarzania. Ponadto ma ono 18 minikomputerów MERA-100 pracujących lokalnie u klientów.

Warszawskie Przedsiębiorstwo Informatyki Przemysłu Budowlanego ETOB
01-244 Warszawa, ul. Józefa Bema 65
Teleks: 813816 ETOB PL
Telefony:
Dyrektor — 32-58-70
Z-ca dyrektora ds. Produkcji — 32-19-30
Z-ca dyrektora ds. Ekonomiczno-Finansowych — 32-68-72
Główny Elektronik — 32-30-46
Projektowanie systemów — 32-58-70
Rozpowszechnianie systemów i obsługa klientów — 32-29-64
Koordynacja produkcji i technologia przetwarzania — 32-58-62
Obsługa informatyczna Ministerstwa BIPMB — 21-015 wew. 200 lub 1174.

Rozwój organizacyjny i sprzętowy powodował także rozwój zastosowań (rys.). W 1953 r. przedsiębiorstwo warszawskie obsługiwało jedno przedsiębiorstwo jednym systemem. W 1978 r. dysponowało ono 30 systemami i obsługiwało 390 przedsiębiorstw. Obecnie WPIPB ETOB dysponuje łącznie 40 systemami informatycznymi i obsługuje 150 przedsiębiorstw. Zmniejszenie się liczby obsługiwanych przedsiębiorstw w ciągu ostatnich pięciu lat wynika z przekazania przedsiębiorstw regionu białostockiego i lubelskiego do ośrodków miejscowych, z zaprzestania prowadzenia bieżącej kontroli inwestycji o szczególnej wadze dla gospodarki narodowej, a także z reorganizacji budownictwa w związku z wprowadzaną reformą gospodarczą.

Do najbardziej rozpowszechnionych systemów informatycznych należą:
— gospodarka materiałowa i obrót towarowy
— gospodarka wyrobami gotowymi
— ewidencja środków trwałych
— zatrudnienie i płace
— gospodarka sprzętowo-transportowa
— planowanie, limitowanie, harmonogramowanie i kosztorysowanie środków produkcji
— obliczenia inżynierskie na potrzeby biur projektów

KAZIMIERZ PAKULSKI
WPIPB ETOB

Poczucie gruntu

Rozmowa z MARIANEM URAZEM, dyrektorem Warszawskiego Przdsiębiorstwa Informatyki Przemysłu Budowlanego ETOB

Redakcja: — Trzydziestolecie warszawskiego przedsiębiorstwa wypadło w okresie niejakiego chaosu polskiej gospodarki. Przy tym cała sieć ETOB została ostatnio zreorganizowana. Czy można zatem mówić o jakimś kończącym się etapie?

Marian Uraz: — Nie, to nie jest etap zamknięty. W początku lat osiemdziesiątych uzupełniliśmy konfigurację posiadanych maszyn, dość sprawnie zresztą pracujących. Sądzimy, że nawet bez znacznych inwestycji w ciągu najbliższych lat możliwa będzie efektywna, satysfakcjonująca użytkowników eksploatacja sprzętu. Chcemy go zresztą niebawem uzupełnić pamięcią półprzewodnikową (256 K) i dużymi dyskami oraz innymi urządzeniami peryferyjnymi. Od ponad dwóch lat prowadzimy zdalne przetwarzanie — u użytkowników jest już osiem końcówek — minikomputerów MERA-100. Ponadto wydajność sprzętu znacznie podwyższyło wykorzystanie systemu GEORGE-3. To wszystko pozwala już dzisiaj niemal w pełni zaspokajać potrzeby budownictwa.

Ostatnie lata stały się dla nas cezurą przede wszystkim w związku z kadrą. A raczej — z jej ubytkiem. Część osób skorzystało z wcześniejszych emerytur, z rent, a także z Uchwały 196 — ci poszli do pracy na swój rachunek. Ale podstawowa kadra została, w tym wiele osób z dużą praktyką — zatem tragedii nie ma.

Dla tych przełomowych lat charakterystyczny jest również proces bezwzględnej weryfikacji eksploatowanych systemów. Użytkownicy zrezygnowali z wielu z nich, uznając, że nie są dostatecznie przydatne. Poza tym niektóre systemy opracowywane były na potrzeby jednostek centralnych; teraz układ się zmienił. A pamiętajmy przy tym, że potrzeby informatyczne budownictwa, w następstwie jego stagnacji, musiały z początku wyraźnie zmaleć; wówczas niewykorzystane były nasze maszyny i ludzie.

Red.: — Ten stan już minął?

M. U.: — Jeszcze nie w pełni. Ale obserwuje się wyraźną poprawę. Wiele przedsiębiorstw budowlanych do tej pory wątpiących w przydatność informatyki, jeszcze w ubiegłym roku uznających jej wykorzystanie za jeszcze jeden zbędny wydatek, zmieniło ostatnio zdanie. Po prostu zaczęło brakować im ludzi. Teraz nasz sprzęt pracuje już na trzy pełne zmiany. To też za sprawą modyfikacji systemów, które staraliśmy się uaktualnić. Reforma gospodarcza zwiększyła — na przykład — znaczenie rozliczeń finansowych (gł. kosztów budownictwa!), a tu bez informatyki ani rusz.

Red.: — Przedsiębiorstwo zajmuje się przede wszystkim systemami ewidencyjnymi?

M. U.: — Ale nie tylko. Złożyliśmy właśnie ofertę generalnemu wykonawcy elektrowni jądrowej w Żarnowcu na kompletną obsługę informatyczną budowy tej elektrowni. Niektóre powstałe w tym celu systemy miałyby być powtarzalne — do wykorzystania przy innych planowanych obecnie budowach tego typu. A nasz sprzęt, jak już powiedziałem, na kilka lat — mniej więcej pięć — wystarczy, by zaspokoić wszystkie potrzeby. W przypadku elektrowni — przetwarzania danych na ich rzecz podejmowałoby się najbliżej położone przedsiębiorstwo ETOB.

Ponadto generalna dyrekcja budowy warszawskiego metra przyjęła naszą ofertę na obsługę informatyczną tej budowy. Wstępne prace już się rozpoczęły. Informatyka miałaby się tu zająć nie tylko problemami ewidencji, ale przede wszystkim — koordynacji prac budowlanych. Na to także wystarczy nam środków technicznych.

Systemy mają być pisane i na ODRĘ, i na przyszłe komputery Jednolitego Systemu.

Red.: Czyli nabytki większych maszyn są jednak planowane?

M. U.: — W najbliższym czasie zamierzamy zlokalizować nasze przedsiębiorstwo w jednym miejscu i do 1985 roku przygotować nową salę do instalacji maszyny RIAD, a najpóźniej w 1987 — zainstalować ją. Być może będzie to już, zapowiadany przez ELWRO, R-34. Oczywiście, na początku będziemy wykorzystywać równolegle i ODRY, i nową maszynę. Emulator opracowany zostanie zapewne wspólnym wysiłkiem zrzeszonych przedsiębiorstw.

Red.: — A co z nie zrzeszonymi?

M. U.: — To w ogóle dziwna sprawa. W Warszawie powstała „czapka" administracyjna. Centrum ETOB, spełniające do 1981 roku funkcje zjednoczenia, zdołało — póki mogło — podporządkować sobie część zakładów podległych przedsiębiorstwom — choćby Lublin (przedsiębiorstwo warszawskie), Kielce (krakowskie), Olsztyn (gdańskie). Powstało w ten sposób jakieś dziwaczne przedsiębiorstwo (w Warszawie mające tylko „czapkę") które oczywiście z nikim nie zamierza się zrzeszać. Na domiar złego w przedsiębiorstwach w Gdańsku, Krakowie i Rzeszowie w trakcie tworzenia się zrzeszenia nastąpiły zmiany na stanowisku dyrektora. Nie było komu podjąć decyzji. Wszyscy dotychczasowi dyrektorzy chcieli konsolidacji. Ale cóż, nie wyszło.

Red.: Jakaś współpraca chyba jednak istnieje?

M. U.: — Tak, oczywiście — ze wszystkimi. Szczególnie z przedsiębiorstwem w Krakowie (drugim trzydziestolatkiem). Sądzę zresztą, że przystąpią do zrzeszenia. Mogą na tym tylko skorzystać.

Red.: — Przedsiębiorstwa ETOB wykonują usługi dla różnych zleceniodawców, nie tylko dla budownictwa?

M. U.: — Usługi wykonywane poza resortem to znikoma mniejszość. Pomagamy ZUS-owi przy przygotowaniu danych na MERZE-9150, współpracujemy trochę z PKO i PKP, obsługujemy niektóre agendy Urzędu Miasta. Ale to naprawdę niewiele. Niemniej przyjmujemy w zasadzie każdego klienta, a ceny na nasze usługi wzrosły nieznacznie, w ciągu ostatnich trzech lat tylko do 25%; było to możliwe dzięki unowocześnieniu parku maszynowego i technologii przetwarzania.

Red.: — Na koniec przedstawmy jeszcze nasze wspólne plany wydawnicze. Zaproponowaliśmy Zrzeszeniu uruchomienie na łamach INFORMATYKI jednokolumnowej rubryki pod jego własną winietą. Czy Zrzeszenie podjęło już decyzję?

M. U.: — Parę dni temu próbowaliśmy rozstrzygnąć tę kwestię. Na razie jednak decyzji nie podjęto. Chcemy najpierw przedstawić Czytelnikom INFORMATYKI materiały (także artykuły problemowe) prezentujące pracę poszczególnych przedsiębiorstw ETOB (jeszcze Kraków i Katowice), a potem dopiero — w zależności od doświadczeń związanych z tymi publikacjami — zadecydować o stałej współpracy z INFORMATYKĄ.

Red.: — Rubryka nie musiałaby się ukazywać regularnie, w każdym numerze. Mogłyby być przerwy. Nawet wówczas byłby to spory doping do zwiększenia przepływu informacji i myśli.

M. U.: — Zapewne. Ale nie spieszmy się. Do tego przedsięwzięcia trzeba się przygotować.

Oprac. Z. GLUZA

Komputery i urządzenia peryferyjne

Komputery, dysponujące dzisiaj bardzo dużymi możliwościami — przy niskich cenach i małych wymiarach, łatwe w stosowaniu zaczynają wpływać na życie codzienne mieszkańców uprzemysłowionego świata. Kolejny krok został zrobiony przez Japonię, gdzie prowadzona jest — z udziałem przedstawicieli rządu, wyższych uczelni i przemysłu — dyskusja o systemach i komputerach, które przekształcą społeczeństwo tego kraju.

We wrześniu 1982 r. odbyła się w Tokio Międzynarodowa Konferencja Systemów Komputerowych Piątej Generacji. W referacie podstawowym przewodniczący tej konferencji, prof. Tohru Moto-oka stwierdził: „W latach dziewięćdziesiątych systemy przetwarzania informacji będą głównymi narzędziami we wszystkich dziedzinach aktywności społecznej (...). Systemy piątej generacji będą musiały zaspokajać nowe potrzeby wywoływane zmianami otoczenia; oczekuje się, że będą one spełniały nie tylko aktywną rolę w pokonywaniu przewidywanych barier rozwoju społecznego, ale także przyczynią się do tego rozwoju".

Nie tylko w Japonii, ale także w USA i Europie, poszukuje się metod ułatwiających korzystanie z maszyn oraz sposobów, które umożliwiłyby komputerowi zrozumienie i interpretację danych wejściowych i ograniczone wnioskowanie, a więc naśladowanie człowieka-specjalisty. Wymaga to uzyskania rezultatów w takich dziedzinach badań, jak: sztuczna ineligencja, systemy podstaw wiedzy (ang. knowledge-based systems), nowe architektury komputerowe, systemy niewrażliwe na uszkodzenia oraz pamięci masowe o dużej gęstości zapisu.

W celu zwiększenia wydajności i niezawodności w wielu systemach stosowana jest różnorodna wieloprocesowa architektura równoległa. Pro-

wadzone są badania nad dwiema nowymi technikami zapisu danych: zapis poprzeczny (ang. vertical recording) i magneto-optyczny — już wkrótce zapewnią większą pojemność pamięci, przy mniejszej objętości i niższej cenie.

Projektowanie nowych, zaawansowanych technologicznie maszyn staje się coraz trudniejsze. Dlatego też do projektowania bardziej skomplikowanych komputerów, układów cyfrowych i systemów stosuje się w przemyśle najnowsze techniki komputerowe. Jedną z kluczowych technik wprowadzonych do systemów projektowania wspomaganych komputerem (ang. computer aided design) i do komputerowych systemów obliczeń inżynierskich jest grafika komputerowa. Technika ta, ze względu na dużą użyteczność reprezentacji graficznej, jest stosowana szeroko także w innych zastosowaniach komputerowych.

Wśród systemów komputerowych najbardziej prestiżową pozycję zajmuje względnie mała grupa wielkich superkomputerów przeznaczonych do obliczeń naukowych, a także powiększająca się wciąż grupa specjalizowanych instalacji handlowo-finansowych (ang. commercial applications). Do tej pory nowy superkomputer pojawiał się co cztery, pięć lat, ale w ubiegłym roku zapowiedziano pięć nowych superkomputerów, po dwa z firm HITACHI i FUJITSU i jeden z CRAY RESEARCH INC.

Komputery przenośne i osobiste, bardzo już rozpowszechnione, umożliwią wkrótce realizację wielofunkcyjnych stanowisk pracy dla dużej grupy pracowników biurowych. Występująca na wszystkich poziomach większych komputerów tendencja do architektury wieloprocesorowej zaczyna przenikać także do tych małych komputerów, będzie się tu nasilała w postaci wyodrębnienia różnorodnych funkcji.

INSTALACJE KOMPUTEROWE PIĄTEJ GENERACJI

Jednym z najambitniejszych w historii projektów komputerowych jest realizowany w Japonii projekt komputerowych systemów piątej generacji. Zakłada się w nim zbudowanie rodziny komputerów i systemów o zupełnie nowych możliwościach: porozumiewanie się z człowiekiem w języku naturalnym, rozumienia wprowadzanych informacji, rozwiązywania problemów drogą myślenia indukcyjnego. Przy realizacji projektu z rządem Japonii współpracują krajowe firmy komputerowe i elektroniczne. Realizacja projektu rozpoczęła się w ubiegłym roku i jest przewidziana na dziesięć lat. Zakłada się jednak, że powinna przynosić także rezultaty pośrednie, m.in. w dziedzinie układów scalonych, obliczeń inżynierskich wspomaganych komputerowo czy technik oprogramowania.

Architektura komputerów piątej generacji odchodzi od klasycznej koncepcji maszyny von Neumanna z jej pojedynczym, sekwencyjnym strumieniem instrukcji, w kierunku nowej architektury — kładącej duży nacisk na przetwarzanie równoległe. Dalsze zwiększenie szybkości pojedynczego procesora jest nieprawdopodobne, stosowane więc będzie powszechnie rozwiązanie polegające na przekazywaniu fragmentów rozwiązywanego problemu do wielu procesorów pracujących w tym samym czasie.

W 1982 r. zrealizowany został pierwszy system, który łączy jednoczesne przetwarzanie różnorodnych instrukcji z przetwarzaniem wielu strumieni danych. Jest to HEP — heterogenous-element processor (procesor o różnorodnych elementach) firmy DENELCOR. W skład systemu wchodzi do 16 procesorów równoległych, każdy z nich może wykonywać jednocześnie — za pomocą równoległych jednostek funkcjonalnych — do siedmiu zadań głównych. Przy wspólnej pracy procesorów równoległych uzyskiwana jest szybkość przetwarzania 160 MIPS (ang. million instructions per second, milionów instrukcji na sekundę).

Większość zapowiedzianych w ubiegłym roku komputerów posiadała choćby niektóre cechy architektury równoległej. W obecnych projektach występują już dziesiątki elementów przetwarzających lub strumieni przetwarzanych. W przyszłości możliwa będzie jednoczesna praca setek, a prawdopodobnie nawet tysięcy procesorów w pojedynczym systemie obliczeniowym.

Ważnym aspektem przetwarzania równoległego jest uzyskiwana w konstrukcjach wieloprocesorowych niewrażliwość na uszkodzenia (ang. fault tolerance). Tylko kilka wieloprocesorowych systemów przetwarzania równoległego zaprojektowano z myślą o uzyskaniu jak największej szybkości obliczeń; inne systemy wykorzystują zwielokrotnienie procesów do zapewnienia ciągłości przetwarzania i do zwiększenia wydajności. Jednym z pierwszych niewrażliwych na uszkodzenia systemów przeznaczonych do pracy w czasie rzeczywistym jest system NonStop firmy TANDEM COMPUTERS.

PAMIĘCI MASOWE

Szybki postęp dokonywany jest również w dziedzinie konstrukcji pamięci magnetycznych. Dyski sztywne, dyski elastyczne, pamięci taśmowe — mają coraz lepsze parametry. Są też coraz tańsze.

Nadal wzrasta gęstość zapisu i powiększa się pojemność pamięci, przy jednoczesnym spadku kosztów zapamiętania pojedynczego bitu. Wciąż jeszcze stosowana jest technika zapisu wzdłużnego, w której cienkie obszary namagnesowane leżą w płaszczyźnie powierzchni nośnika. Traci ona jednak na znaczeniu na skutek fizycznych ograniczeń w procesie zapisu. Na jej miejsce wejdą zapewne dwie nowe techniki: magnetycznego zapisu poprzecznego (ang. vertical lub perpendicular magnetic recording) i laserowego zapisu magneto-optycznego. Techniki te umożliwią znaczne zwiększenie gęstości zapisu w pamięciach wymazywalnych. Przy użyciu nośników stosowanych obecnie w dyskach sztywnych, dyskach elastycznych i pamięciach taśmowych (wykorzystujących tlenki żelazawe), nowe techniki umożliwiają 10-krotny, a prawdopodobnie nawet 50-krotny wzrost pojemności pamięci.

Zapis poprzeczny potrzebuje nośnika, w którym niewielkie regiony magnetyczne ustawiają się, po namagnesowaniu, w kierunku poprzecznym do jego powierzchni. Regiony te powinny być gęsto upakowane, a ich stopień namagnesowania powinien umożliwiać jednoznaczne określenie kierunku namagnesowania podczas odczytu. Chociaż pierwsze badania i próby z zapisem poprzecznym były prowadzone we Francji, Włoszech i USA, to najpoważniejsze rezultaty uzyskano w Japonii pod kierownictwem prof. Shun-ichi Iwasaki z Uniwersytetu Tohoku w Sondai.

Prace nad jednostkami pamięci na dyskach sztywnych i elastycznych wykorzystującymi zapis poprzeczny prowadzi się w Japonii, USA i Europie. W Japonii nad tymi wyrobami pracują firmy FUJITSU, HITACHI, MATSUSHITA ELECTRIC INDUSTRIAL, NIPPON ELECTRIC, NIPPON TELEGRAPH AND TELEPHONE, SONY i TOSHIBA. Ponieważ zapis poprzeczny może być użyty do zapisu dźwięku i obrazu, przypuszcza się, że niektóre firmy japońskie prowadzą prace nad urządzeniami do tych celów. Wysoka gęstość zapisu, uzyskiwana przy zapisie poprzecznym, umożliwi wysoką jakość rejestracji i odtwarzania. Pojawi się być może płyta wizyjna (o bardzo dużej ostrości obrazu) z cyfrowym, stereofonicznym zapisem dźwięku, zapewniająca uzyskanie lepszej wierności obrazu niż w obrazie filmowym.

W Paryżu, w firmie CII-HONEYWEL BULL, prowadzone są prace nad dyskiem sztywnym wykorzystującym zapis poprzeczny, a w firmie OLIVETTI we Włoszech — nad dyskiem elastycznym. W USA, oprócz głównych producentów sprzętu komputerowego i peryferyjnego, prace nad zapisem poprzecznym prowadzą też trzy nowe firmy: VERTIMAG SYSTEMS zaprezentowała dysk elastyczny i jednostkę dyskową 5 1/4 cala, a LANX i APPLIED INFORMATION MEMORIES opracowują dysk sztywny. Firma ANELVE z Tokio rozesłała próbne egzemplarze 5 1/4-calowego dysku z nośnikiem kobaltowo-chromowym przeznaczonego do zapisu poprzecznego. Pierwsze jednostki dyskowe wykorzystujące zapis poprzeczny są dostępne na rynku od początku bieżącego roku.

W październiku 1982 r. pojawiły się na rynku urządzenia do przechowy-

wania informacji cyfrowej na niewymazywalnych optycznych dyskach laserowych. Są one przeznaczone do zastosowań archiwalnych. Osiągnięto także znaczący postęp w konstrukcji wymazywalnych laserowych dysków magneto-optycznych, które mogą być wykorzystywane do zapisu danych cyfrowych lub obrazu. Jako nośnik stosowane są cienkowarstwowe materiały amorficzne. Do zapisu w tych dyskach wykorzystuje się zjawisko termomagnetyczne, a do odczytu — zjawisko Kerra. W ośrodku badawczym firmy XEROX uzyskano obiecujące rezultaty stosując czterowarstwowy, cienkowarstwowy dysk amorficzny. Jedną z japońskich zapowiedzi jest prototyp wymazywalnej płyty wizyjnej z magneto-optycznym zapisem laserowym na nośniku dwuwarstwowym — firmy NHK.

W dziedzinie konwencjonalnych dysków magentycznych wprowadzono w ubiegłym roku ponad 50 nowych wyrobów. Producenci dążą nadal do zmniejszania wymiarów dysków — do średnicy 5 1/4 cala, a nawet do 3—3,9 cala. Małe jednostki dyskowe są potrzebne do nowych małych komputerów. Pojemność dysków będzie przy tym rosła, co umożliwi wzrost mocy obliczeniowej komputerów, mimo zmniejszenia ich rozmiarów.

AUTOMATYZACJA PROJEKTOWANIA

Stopień złożoności, jaki został osiągnięty w systemach komputerowych i składających się na nie układach scalonych, spowodował konieczność użycia samych komputerów do pomocy w ich projektowaniu. Trwa więc okres świetnej koniunktury na nowe systemy automatyzacji projektowania. Spadek cen komputerów, zmniejszenie ich wymiarów i wzrost możliwości obliczeniowych powodują szerszą dostępność systemów automatyzacji projektowania.

Nowe systemy automatyzacji projektowania są połączeniem sprzętu i oprogramowania. W ich skład wchodzą inteligentne inżynierskie stanowiska robocze, mini- i mikrokomputery 16- i 32-bitowe do wykonywania obliczeń oraz programy do realizacji takich zadań, jak: symulacja logiczna czy analiza czasowa układów scalonych. W inżynierskich stanowiskach roboczych wykorzystywane są najnowsze osiągnięcia grafiki komputerowej i techniki mikroprocesorowej, a także nowości przemysłu pamięci półprzewodnikowych i dysków.

Nowe systemy umożliwiają interakcyjne projektowanie logiczne i usuwanie błędów projektowych za pomocą graficznych stanowisk pracy. Systemy te służą do projektowania układów scalonych i urządzeń na płytkach; wykorzystują do tego celu wspólną bazę danych. Część systemu, przeznaczona do projektowania logicznego, umożliwia szkicowanie i projektowanie oraz analizę logiczną za pomocą programów symulacyjnych i procedury kontroli poprawności logicznej (umożliwiającej np. sprawdzenie zależności czasowych).

Wśród wielu systemów automatyzacji projektowania wprowadzonych na rynek w ubiegłym roku, jest kilka, w których wykorzystano obecny stan techniki. Odpowiadają one swym poziomem wiedzy inżyniera elektronika. Systemy te, gdy tylko w użyciu znajdzie się większa ich liczba, mogą spowodować rewolucję w projektowaniu. Liczba nowych firm specjalizujących się w wytwarzaniu takich systemów świadczy o pojawieniu się nowej gałęzi przemysłu — zautomatyzowanych systemów projektowania i wytwarzania elementów elektronicznych.

REWOLUCJA GRAFICZNA

Rewolucja sprzętu grafiki komputerowej właśnie się zaczyna. W 1982 roku wprowadzono na rynek wiele nowych wyrobów, takich jak: terminale, specjalizowane procesory graficzne, pisaki X-Y (plotery), drukarki. A w najbliższych latach asortyment ten jeszcze się poszerzy.

Jedyną barierą hamującą w ostatnich latach stosowanie grafiki w systemach komputerowych był znaczny koszt sprzętu wysokiej jakości i niewystarczające środki dla oprogramowania. Obydwa te ograniczenia są teraz szybko łagodzone — koszt sprzętu graficznego szybko maleje zaś tworzenie oprogramowania ułatwiają wydajniejsze systemy projektowania oprogramowania i wprowadzane standardy oprogramowania graficznego, umieszczające podstawowe funkcje w układach scalonych. Zastosowanie grafiki i koloru ułatwi w najbliższych latach wzajemne oddziaływanie na siebie człowieka i komputera.

DUŻE KOMPUTERY

Duże i szybkie komputery przeznaczone do przetwarzania danych nadal znajdują szerokie zastosowanie. Wielu producentów zapowiedziało w ubiegłym roku wprowadzenie nowych procesorów o większej szybkości wykonywania operacji. Firma IBM zapowiedziała model K, drugi w serii 3081 o szybkości ok. 14 MIPS. Szybkość ta zapewnia wzrost wydajności o 40% w porównaniu z pierwszym modelem (D) tej serii. We wrześniu 1982 r. IBM wprowadził procesor 3084 o szybkości, 1,9 raza większej od szybkości modelu K. Parametry tego procesora są porównywalne z parametrami procesorów firm AMDAHL, FUJITSU i HITACHI. Produkujące sprzęt niekompatybilny ze sprzętem IBM, firmy CDC, HONEYWELL i SPERRY UNIVAC także uzupełniły swoje serie maszyn nowymi, największymi modelami.

Szybkość dużych komputerów nieustannie wzrasta, tak jak i liczba instrukcji na sekundę w przeliczeniu na 1 dol. ceny sprzętu. Nadal rozszerzany jest zakres stosowania dużych komputerów, m.in. w następstwie rozwoju sieci komputerowych. Wydaje się, że tradycyjne przetwarzanie danych w zastosowaniach handlowych i finansowych będzie nadal wykonywane na dużych komputerach. Szybki wzrost zapotrzebowania na dostęp do baz danych nastąpi natomiast w przypadku różnorodnych systemów przetwarzania informacji w biurach, fabrykach, przy obliczeniach indywidualnych czy przetwarzaniu transakcji.

Oprócz dużych komputerów, do przetwarzania danych przeznaczone są także superkomputery. Zapotrzebowanie na nie nieustannie wzrasta. Do dwóch producentów amerykańskich, firm CDC i CRAY RESEARCH, dołączyły w ubiegłym roku dwie wspomniane już firmy japońskie: FUJITSU i HITACHI. Firma CRAY zapowiedziała model X-MP, szybszy od 2,5 do 5 razy od modelu CRAY 1-S, mogącego wykonać 140 mln operacji zmiennoprzecinkowych na sekundę. Superkomputery FUJITSU — VP-100 i VP-200 działają odpowiednio prawie 2 i ok. 3,5 raza szybciej niż CRAY 1-S, a HITACHI — S-810/10 i S-810/20 odpowiednio 2,5 i 4,5 razy szybciej. Szybkość superkomputera CYBER 205 firmy CDC, wprowadzonego na rynek kilka lat temu, znajduje się na tym samym poziomie.

Na całym świecie instaluje się obecnie ponad 50 superkomputerów, przy czym 85% z nich pochodzi z firmy CRAY. Według szacunków producentów, rynek światowy wchłonie w najbliższych kilku latach kilkaset takich systemów.

MAŁE KOMPUTERY

Sporymi możliwościami obliczeniowymi dysponują obecnie komputery stołowe (ang. desk-top computer) i komputery przenośne (ang. portable computer). Widoczna jest też tendencja umieszczania coraz większej mocy obliczeniowej w coraz mniejszych obudowach. Komputery, niewiele większe od klawiatury z dołączonymi do nich małymi ekranami, umieszczane są w wygodnych obudowach stołowych lub przystosowane do łatwego przenoszenia. Wiele spośród nich dysponuje możliwościami obliczeniowymi większych komputerów sprzed niewielu lat — wykonują do 1 MIPS, a ich szybkość nadal wzrasta. Możliwości komputerów osobistych będą się powiększały na skutek wprowadzania nowych technologii wytwarzania płaskich ekranów (ang. flat-panel display) oraz nowych konstrukcji zwartych pamięci masowych, takich jak: pamięci pęcherzykowe i 2- lub 3-calowe jednostki dyskowe z zapisem poprzecznym.

Dobry przykład możliwości komputera osobistego daje COMPASS firmy GRID SYSTEMS. Komputer ten waży mniej niż 10 funtów i mieści się w płaskim neseserze. W jego skład wchodzą dwa procesory 16-bitowe, 256 K bajtów pamięci RAM, pamięć pęcherzykowa, płaski adresowany bitowo

ekran elektroluminescencyjny o bursztynowej barwie świecenia i wymiarach 3 1/2 × 4 3/4 cala, pełna klawiatura, modem i telefon. Przy cenie 8150 dol. nie znajdzie pewnie zbyt wielu nabywców, ale jest zapowiedzią konstrukcji przeznaczonych do wydajnego przetwarzania zorientowanego graficznie.

Długoplanowe badania nad rozpoznaniem głosu odegrają w przyszłości dużą rolę w poprawianiu warunków ergonomicznych różnych terminali. Zostaną opracowane konstrukcje łączące wprowadzanie danych i udzielanie pewnych odpowiedzi głosem, z wejściem z klawiatury, wejściem graficznym i kolorowym monitorem ekranowym o dużej rozdzielczości.

Komunikaty przekazywane głosem i wskaźniki optyczne są uważane za nietrwałe, ponieważ przekazywana za ich pośrednictwem informacja nie może być „wyniesiona" z pomieszczenia, w którym jest terminal i przekazana osobie, która nie ma terminala. Ludzie nie są jeszcze przygotowani do rezygnacji z trwałej kopii papierowej. Z tej sytuacji zdają sobie sprawę producenci drukarek komputerowych. W ubiegłym roku zapowiedzieli oni wprowadzenie wielu nowych konstrukcji. Są wśród nich nowe drukarki laserowe, drukarki natryskowe (ang. ink-jet) i wiele drukarek barwnych z różnymi rodzajami mechanizmu drukującego. Wszystkie konstrukcje charakteryzuje duża niezawodność i stosunkowo niska cena.

ROBOTY

Techniki komputerowe i telekomunikacyjne znalazły ostatnio szerokie zastosowanie w przemysłowych systemach automatyzacji. Coraz większy nacisk kładzie się przy tym na stosowanie robotów. Wszystkie społeczeństwa przemysłowe, podejmując wysiłki w celu zwiększenia wydajności produkcji, kierują swoją uwagę właśnie na roboty i inne techniki automatyzacji. Przoduje w tej dziedzinie Japonia, która sprowadziła z USA niektóre wczesne konstrukcje. Roboty te, po udoskonaleniu, zostały zastosowane w wielu przedsiębiorstwach japońskich na niespotykaną gdzie indziej skalę. Jedynie firma HITACHI podjęła przed dziesięciu laty samodzielne badania w dziedzinie robotyki.

Przepływ technologii został obecnie odwrócony. Kilka wielkich firm amerykańskich musiało kupić licencje na roboty zaprojektowane w Japonii i Europie. Na przykład — GENERAL ELECTRIC jest licencjobiorcą firmy HITACHI, kupił też licencję na wiele robotów od włoskiej firmy DEA.

Występuje tendencja do zmniejszania wymiarów robotów i obniżania ceny, aby umożliwić szersze ich stosowanie. Firma IBM wprowadziła na rynek łatwy do zamontowania robot, wytwarzany przez SANKYO SEIKI MANUFACTURING z Tokio. Może on być programowany za pomocą PERSONAL COMPUTER. Następne tego typu wyroby z Japonii i Europy Zachodniej już wkrótce pojawią się na rynku.

Tymczasem, znów w Japonii, firma MITSUBISHI ELECTRIC wprowadziła niewielki „robot-dziecko", z myślą o kształceniu programistów-robotów; może on jednak znaleźć zwykłe zastosowania robocze.

W Massachusetts Institute of Technology podjęto wieloletnie badania, prowadzone z finansową pomocą rządu, w celu osiągnięcia tzw. skoku kwantowego w robotyce.

Opracował JACEK WALAS
na podstawie
ELECTRONICS z 20.10.1982

Firmy amerykańskie

Poniższe opracowanie przedstawia czołówkę amerykańskich firm komputerowych w pierwszej połowie 1982 roku. Warto śledzić sytuację tego rynku, jako że stanowi ona wykładnię aktualnych tendencji informatyki światowej.

W tabeli 1 przytoczona jest lista dziesięciu największych firm komputerowych — według dochodów uzyskanych w dziedzinie informatyki w pierwszej połowie 1982 r. Jak widać, w ciągu roku nie nastąpiły istotne zmiany. BURROUGHS awansował z 5 miejsca na 3, dzięki przejęciu firmy MEMOREX. Największy procentowy przyrost wykazała firma STORAGE TECHNOLOGY, awansując po raz pierwszy do czołowej dziesiątki. Dalej: WANG LABORATORIES (przetwarzanie tekstów), HEWLETT-PACKARD (zamiana miejsca z HONEYWELL), DEC i IBM.

W porównaniu z sytuacją z 1980 r.[1]) widać natomiast wyraźny awans firmy DEC, która już w pierwszej połowie 1981 roku zajęła drugie miejsce na liście, będąc czwartą w 1980 r. i szóstą w roku 1979. Ponadto z pierwszej dziesiątki wypadły firmy XEROX i MEMOREX. Pierwsza zajmowała w

[1]) Jan Ryżko: Przemysł komputerowy USA w roku 1980. INFORMATYKA nr 2—3/1982, s. 34—35

Tabela 1. Największe amerykańskie firmy komputerowe wg dochodów w I połowie 1982 r.

Lp.	Pozycja w I półroczu 1981	Firma	Dochód (mln dol.)	Procent wzrostu
1	1	IBM	13 229,0	14,5
2	2	DEC	2075,6	16,5
3	5	Burroughs	1762,1	3,0
4	3	Control Data	1565,3	5,4
5	4	Sperry-Univac	1425,2	—2,5
6	6	NCR	1350,1	2,5
7	8	Hewlett-Packard	1063,7	27,0
8	7	Honeywell	876,0	2,7
9	9	Wang Laboratories	648,5	30,2
10	12	Storage Technology	516,6	34,3

Tabela 2. Amerykańskie firmy komputerowe o największym procentowym wzroście dochodów w I połowie 1982 r.

Lp.	Firma	Procent wzrostu	Dochód w I połowie 1981 r.	Pozycje w I połowie 1981 r.
1	Apple Computer	61,5	169,7	25
2	Tandy (Radio Shack)	59,0	204,8	21
3	Storage Technology	34,3	384,6	12
4	Wang Laboratories	30,2	498,2	9
5	Hewlett-Packard	27,0	837,4	8
6	Prime Computer	20,7	172,6	24
7	Automatic Data Processing	18,9	296,0	18
8	DEC	16,5	1781,4	2
9	Tektronix	14,9	162,6	28
10	IBM	14,5	11553,0	1

1982 r. 11 miejsce, a druga — jak już wspomniano — została wchłonięta przez BORROUGHS. Na ich miejsce weszły WANG LABORATORIES i STORAGE TECHNOLOGY.

W tabeli 2, w której zestawione są firmy komputerowe według procentowego wzrostu dochodów w stosunku do pierwszej połowy roku 1981 znalazła się połowa firm z pierwszej dziesiątki, przy czym kolejność jest niejako odwrócona. Pierwsza występuje tutaj STORAGE TECHNOLOGY (pozycja 3), będąca ostatnią w tabeli 1, a ostatnią IMB — tam pierwsza. Najwyższy procentowy wzrost wykazują APPLE COMPUTER i TANDY, które obecnie są w trzeciej dziesiątce firm komputerowych, lecz ciągle awansują.

Należy pamiętać, że podane wskaźniki wzrostu dotyczą okresu, gdy dochód narodowy w USA (uwzględniając inflację) spadł o 2%. Łącznie 100 czołowych amerykańskich firm informatycznych osiągnęło dochód z tej dziedziny w wysokości prawie 75 mld dol., co oznacza średnio 12% wzrostu w stosunku do roku 1981, kiedy dynamika rozwoju była wyższa (wówczas firmy APPLE i TANDY wykazywały wskaźniki wzrostu wyższe od 100%).

Jeśli uwzględnić inflację, rzeczywisty wzrost dochodów trzydziestu czołowych firm informatycznych wyniósł w pierwszej połowie 1982 r. tylko 5,3%. Nominalnie wyniki tych firm są ponad dwukrotnie korzystniejsze od wskaźników dla całej gospodarki USA.

Tylko dwie firmy spośród tych, które występowały w pierwszej trzydziestce w latach 1981—1982, miały wskaźniki rozwoju równe lub gorsze od średniego dla całej gospodarki, a sześć osiągnęło wzrost większy od 20%.

Nową firmą, jaka pojawiła się w pierwszej trzydziestce w roku 1982 jest MOTOROLA (miejsce 23), która wchłonęła firmę FOUR PHASE. Awansowała z 47 pozycji.

IBM swój stosunkowo wysoki wskaźnik wzrostu zawdzięcza ożywionemu rynkowi systemów 308X i towarzyszących urządzeń peryferyjnych.

Opracował JAN RYŻKO
na podstawie
COMPUTER DECISIONS,
styczeń 1983

Kierunki rozwoju oprogramowania w fizyce wielkich energii

W dniach 4—6 października ub.r. w Genewie odbyło się zorganizowane przez CERN — Europejską Organizację ds. Badań Jądrowych, robocze spotkanie na temat oprogramowania stosowanego w fizyce wielkich energii.

W obliczeniach związanych z zagadnieniami fizyki wielkich energii występuje bardzo duża liczba danych i często złożone, zmieniające się programy analizy tych danych. Na ogół we współpracy uczestniczy duża liczba laboratoriów, a także laboratoria nie mające wszystkich możliwości eksperymentalnych, lecz jedynie analizy danych. Ośrodki komputerowe dysponują bardzo zróżnicowanym sprzętem i oprogramowaniem, co utrudnia szerszą wymianę wyników.

Stopień znajomości informatyki w środowisku fizyków jest bardzo różny: od pełnej kompetencji specjalistów znających bardzo dobrze wszystkie problemy obliczeniowe związane z eksperymentem — do fragmentarycznej wiedzy przypadkowych użytkowników komputera. Sytuację w tej dziedzinie kształtują rozmaite czynniki, także psychologiczne, np. dążenie do śledzenia eksperymentu od początku do końca, niechęć do dzielenia się wynikami z kimś innym itp. Poza fazą samego eksperymentu, z jego nadzorowaniem itd., obliczenia są wykonywane zwykle na dużym komputerze (np. IBM 370, CDC 6000) z wykorzystaniem programów napisanych na ogół w FORTRANIE.

W ostatnich latach pojawiły się dwa nowe elementy w tej dziedzinie. Pierwszy — to budowa kosztownych, unikalnych urządzeń, jak np. wielki akcelerator LEP [1]) w Genewie, wymuszająca nowe podejście do projektowania, organizacji i przeprowadzania eksperymentów, a więc także — związanych z nimi obliczeń. Drugi — to świadomość zmian w samej informatyce i dziedzinach pokrewnych. Zmiany te coraz częściej są dostrzegane nawet w życiu codziennym. Z tego względu fizycy zaczynają odczuwać niepokój, czy środki, którymi dysponują, są adekwatne do potrzeb i do ogólnych wymagań. Stąd też zapewne hasło: „Dokąd zmierzamy?", pod którym odbywało się to spotkanie.

W Genewie dyskutowano o stylu (referat Michaela Jacksona) i językach programowania, o nowych narzędziach programistycznych i możliwościach technicznych (urządzenia graficzne, osobiste stanowiska robocze). Przedstawiono metody programowania stosowane w innych dziedzinach np. w brytyjskim przemyśle naftowym, Europejskiej Agencji Kosmicznej ESA, francuskim przemyśle lotniczym, meteorologii, a także — w niemieckim ośrodku badawczym DESY koło Hamburga. Kilka referatów dotyczyło sieci komputerowych, zarówno lokalnych jak i krajowych oraz międzynarodowych.

1) LEP — Large Electron — Positron Collider

Szczególnie interesujące wydają się trzy wątki: spór o języki programowania, zastosowania baz danych oraz problemy związane z sieciami komputerowymi.

SPÓR O JĘZYK PROGRAMOWANIA

Tej tematyki dotyczyły dwa referaty — o ADZIE oraz o FORTRANIE i jego następcach, za które uważa się FORTRAN 77 i FORTRAN 8x. Modyfikacje FORTRANU są dokonywane w kierunku jego strukturalizacji i standaryzacji. FORTRAN 77 zawiera m.in. pełne instrukcje warunkowe i rozszerzone pętle. Mniej znany w Polsce, dopiero tworzony przez grupę X3J3 ANSI, FORTRAN 8x wprowadza m.in. większy stopień modularyzacji programów, rozszerzania operacji na tablicach do przetwarzania wektorowego, rekursję i dynamiczną allokację lokalnych tablic. Język ADA jest znany w kraju m.in. dzięki publikacjom w INFORMATYCE.

Polemika na ten temat była bardzo żywa i rozrosła się ponad miarę w dyskusji końcowej, służącej ogólnemu podsumowaniu spotkania.

Charakterystyczne jest uporczywe trzymanie się FORTRANU przez część środowiska i niechęć do wszelkich zmian w tej dziedzinie. Jak powiedział jeden z dyskutantów, D. Williams (CERN): „fizycy będą pisali programy w dowolnym języku tak długo, jak długo będzie on nazywał się FORTRAN". Zwolennicy tego języka przytaczają szereg argumentów na jego korzyść, jak np. potrzeba użycia języka ukierunkowanego maszynowo (ze względu na operacje bitowe na danych generowanych przez detektory w czasie eksperymentów), zapewnienie wydajnego generowania kodu przez FORTRAN, olbrzymie zasoby zmagazynowane w istniejących bibliotekach i pakietach fortranowskich (za sto lat wciąż mamy mieć FORTRAN?) trudności w uczeniu się nowych, złożonych języków. Podkreślano też łatwość przenoszenia programów napisanych w FORTRANIE.

Z tendencyjnie ułożonej tabeli (M. Metcalf CERN) wynika, że najlepszy jest FORTRAN 8x, co bierze się m.in. stąd, że arytmetyka zespolona, operacje na tablicach, duża liczba funkcji matematycznych i inne możliwości nie są standardowo dostępne w ADZIE, a będą dostępne w FORTRANIE 8x. Nie zauważa się ogólniejszych możliwości ADY polegających na konstrukcji pakietów i ich standaryzacji. Z dużym lekceważeniem mówiono o PASCALU (U. Wambach), stwierdzając, że próby zaadoptowania go w CERN-ie od 1976 roku nie przyniosły pozytywnych rezultatów. ADĘ krytykowano za rzekomą złożoność oraz zasłanianie informacji („Dane mogą być tak schowane przed użytkownikiem, że nie będzie on zdolny ich odnaleźć" — J. Zoll, CERN). Powtarzano też opinię Dijkstry o złej definicji języka, choć trudno jest znaleźć prostszy język do celów, dla których utworzono ADĘ. Jej zwolennicy wskazywali na znaczenie całego przedsięwzięcia i na konieczność uwolnienia się od FORTRANU. Kompilator ADY nie będzie wcale bardziej złożony niż kompilator FORTRANU 8x, a efektywność jest ważniejsza przy tworzeniu i modyfikowaniu programu niż jego wykonywaniu. Jeśli mówimy, że FORTRAN 77 jest łatwiejszy do nauczenia od ADY, czy podświadomie nie myślimy o użytkownikach wychowanych na FORTRANIE? (R. Bock, CERN).

Wydaje się, iż środowisko fizyków nie dostrzega, że ADA nie jest jeszcze jednym ze słabo rozpowszechnionych języków wyrosłych na gruncie uniwersyteckim. ADĘ tworzy, będzie ją rozpowszechniał i dyktował standardy, największy w świecie użytkownik oprogramowania — amerykański Departament Obrony — wraz ze swoimi pieniędzmi i możliwościami. FORTRAN będzie istniał jeszcze długo, ale wiele osób będzie zapewne zaskoczonych tym, jak prędko ADA wraz z oprogramowaniem dodatkowym (jak biblioteki programów i pakiety standardowe) może zdominować rynek.

Ogólną, trochę niejasną sytuację charakteryzują dwie uwagi wypowiedziane w dyskusji końcowej:
— Jak fizycy mogą pisać dobre programy według prawidłowych wzorców, jeśli na co dzień mają do czynienia z najgorszym oprogramowaniem, jakie kiedykolwiek stworzono — z systemem operacyjnym IBM? (Ch. Curran, CERN).
— Wydaje się, że sytuację można opisać następującym stwierdzeniem: fizycy nie wiedzą jak programować, zawodowcy nie wiedzą co programować (J. Zoll, CERN).

BAZY DANYCH

Stan rzeczy w niektórych fizycznych zastosowaniach baz danych odzwierciedla negatywna odpowiedź (K. G. Jeffery, W. Brytania) na pytanie (R. Żelazny, Swierk, Polska): „Czy zna pan rzeczywiste implementacje systemów zarządzania bazą danych do gromadzenia i przetwarzania danych doświadczalnych?" W czasie spotkania przedstawiono jednak kilka interesujących systemów i zastosowań baz danych.

R. P. Worden (Logica, W. Brytania) zaprezentował relacyjny system zarządzania bazą danych RAPPORT, który może być wykorzystywany niezależnie, ale stanowi też część szerszych systemów firmy: Fortran Program Analyzer, FPA i Test Coverage Monitor, TCM, które ułatwiają pracę z dużymi programami napisanymi w FORTRANIE. Pierwszy z nich analizuje program i jego strukturę, drugi — ułatwia przeprowadzanie testów i bada stopień przetestowania poszczególnych części programu. W obu przypadkach informacje są zapisywane w bazie danych, a nie wyprowadzane w postaci bardzo długich wydruków. Użytkownik wybiera te informacje, które go w danej chwili interesują. Tworzony jest także zintegrowany system Static Analysis Package z wbudowanymi obydwoma narzędziami, na zamówienie Europejskiej Agencji Kosmicznej.

Inny system o prowizorycznej nazwie CANTOR, opracowywany w Szwedzkim Instytucie Badawczym Obrony Narodowej, służy do analizy wielkich zbiorów danych o złożonej strukturze informacji. Jego podstawowym elementem jest relacyjny system zarządzania, umożliwiający oszczędne pamiętanie danych i szybkie udzielanie odpowiedzi na pytania. Język zapytań jest oparty na teorii mnogości i przeznaczony przede wszystkim dla naukowców. Do systemu można wbudowywać nowe podsystemy z typowymi udogodnieniami bazy danych. Jak się wydaje, system oparto na solidnych podstawach teoretycznych.

Jeszcze inne zastosowanie znalazł system zarządzania bazą danych w badaniach fuzji magnetycznej, prowadzonych w Ecole Polytechnique Federale w Lozannie. Opracowano tzw. kod ERATO oparty na koncepcji systemu OLYMPUS. Użytkownikowi podaje się szkielet programu badającego stabilność jednego z typów równań, a on sam uzupełnia program, składający się z kilku bloków funkcjonalnych. Każdy z bloków współpracuje z bazą danych.

Informacje o zastosowaniach baz danych przedstawiono w opisie wielu innych systemów. Wydaje się, że systemy, których centralnym elementem jest baza danych, będą — w różnych zastosowaniach — coraz bardziej rozpowszechnione i dotrą również do środowiska fizyków. Już obecnie, w ramach przygotowań do prac związanych z przedsięwzięciem LEP, jedna z grup bada możliwości w tej dziedzinie.

SIECI KOMPUTEROWE

Na spotkaniu przedstawiono przegląd metod i technologii stosowanych w sieciach lokalnych i podano przykład takiej sieci (HMINET-2 w Hahn-Meitner-Institut w Berlinie Zachodnim). Zaprezentowano także stopień realizacji planów dotyczących rozwoju sieci w Wielkiej Brytanii, w skali całego kraju oraz ich połączeń z innymi sieciami.

Problem sieci komputerowych jest dla fizyków o tyle ważny, że środowisko to stoi przed koniecznością zintensyfikowania prac w celu polepszenia komunikacji zarówno w ramach pojedynczych laboratoriów i ośrodków badawczych, jak i pomiędzy nimi. Niektóre ośrodki mają już własne sieci lokalne, a inne korzystają z dostępu do sieci obejmujących różne fragmenty kraju czy świata. Przykładem może być właśnie lokalna sieć HMINET-2. Ma ona strukturę gwiazdową i jest oparta na zalecanym przez CCITT standardzie X.25. Finansujące większość prac Ministerstwo Rozwoju Technologii RFN (BMFT) traktuje ją jako projekt pilotujący. Wykorzystuje się w niej komputery: SIEMENS 7760, 7880 i 330, VAX 11/780, PDP-11, LSI-11. Sieć ma połączenie z zachodnioniemiecką siecią DATEX-P przez jej węzeł berliński. Ta sieć z kolei, od 1981 roku ma połączenia z sieciami TRANSPAC, TELENET, DATAPAC i IPSS.

W referatach przeglądowych dotyczących sieci lokalnych, zarówno P. Zafiropulo, jak i K. Kümmerle (Szwajcaria) jako najlepszą wśród technik stosowanych w tych sieciach, wskazali rozproszony system pierścieniowy z metodą dostępu oznakowego (ang. token access).

Według B. J. Charlesa, w Wielkiej Brytanii istnieją obecnie dwie sieci o szerokim zasięgu: publiczna SWITCHSTREAM-1 lub PSS, która ma obecnie 25 połączeń typu X.25 z 25 ośrodkami i prywatna sieć (także oparta na normie X.25) Science and Engineering Research Council — SERCNET — z ok. 100 połączeniami dla 55 ośrodków. Sieci te mają połączenie między sobą, a w końcu roku 1982 miały być możliwe międzynarodowe połączenia z 17 krajami (m.in. z USA, Hong Kongiem, Singapurem, a także z siecią EURONET).

W dyskusji wskazywano na brak standaryzacji technologii i urządzeń (wiele prac ISO jest jeszcze w toku), a także na problemy związane z rozwojem sieci, jak np. zmianą warunków rozpowszechniania oprogramowania, udostępnianego przez sieć.

* * *

Trudno obecnie ocenić jaki wpływ wywrą takie spotkania i dyskusje na praktykę programowania w fizyce wielkich energii. Okaże się to zapewne już wkrótce podczas prac związanych z budową akceleratora LEP. Z powyższego przeglądu wynika, że w tej dziedzinie istnieje znaczne zapotrzebowanie na nowoczesne metody programowania, a badania fizyczne mogą być także czynnikiem stymulującym rozwój tych metod.

PIOTR STRZAŁKOWSKI

Terminologia FORTRANU czasu rzeczywistego

W projekcie normy FORTRANU czasu rzeczywistego, której druk rozpoczynamy w bieżącym numerze, najważniejsze pojęcia zostały dość starannie zdefiniowane. Poniżej podajemy spolszczone definicje, jak również polskie odpowiedniki poszczególnych terminów angielskich.

Terminy zaznaczone rozstrzelonym drukiem są wyjaśnione w innych miejscach „słowniczka". Definicje oznaczone gwiazdką (*) pochodzą z normy ISO 2382/X.

CZAS BEZWZGLĘDNY (absolute time) — pełne określenie punktu doby i daty.

CZAS WZGLĘDNY (relative time) — przyrost lub różnica czasu.

INICJOWANIE (initiation) — działanie podjęte przez system nadzorczy w celu rozpoczęcia wykonania zadania od jego pierwszej instrukcji czynnej.

NAŁOŻENIE WYKONAŃ (overrun) — sytuacja polegająca na spełnieniu warunku inicjowania zadania, gdy zadanie wykonuje się jeszcze wskutek uprzedniego zainicjowania.

NIEOBECNE (non-existent) — jeden z określonych (formalnych) stanów zadania. Zadanie nieobecne jest nieznane systemowi nadzorczemu.

OBLICZENIE (computation) — zbiór operacji zastosowanych do zbioru danych, w zakresie dostępnym dla programu użytkowego.

OCZEKUJĄCE (pending) — jeden z określonych stanów zadania. Zadanie niezałatwione jest związane ze zdarzeniem lub czasem, tak że przy spełnieniu warunku przechodzi do stanu wykonywane i zostaje zainicjowane.

OPERACJA (operation) — deterministyczna zasada generacji skończonego zbioru danych z innego skończonego zbioru danych.

PLIK (file) — zestaw pokrewnych rekordów uważany za jednostkę. W niniejszej normie uważa się, że rekordy mają ustaloną długość. Sposób pamiętania rekordów i dostęp do rekordów są niezależne od ich wewnętrznej postaci.

PODSTAWOWE JEDNOSTKI ZEGARA (basic clock units) — jednostki podstawowego bloku systemowego zegara czasu rzeczywistego, dostępne dla programu użytkowego.

PROCESOR (processor) — system przetwarzania danych łącznie z mechanizmem, za pomocą którego programy są przekształcane do wykorzystania w tym systemie (ANSI X3.9-1978).

PROCESOR WIRTUALNY (virtual processor) — środowisko, w którym zadanie może być wykonywane od zainicjowania do zakończenia, bez uwzględnienia dostępności zasobów, zarządzanych przez procesor, a nie program użytkowy. Konkretna realizacja obejmuje odwzorowanie zbioru procesorów wirtualnych na zbiór procesorów rzeczywistych. Odwzorowanie to jest zależne od procesora.

PROGRAM, SYSTEM NADZORCZY (executable routine, system) — część procesora, która wspomaga realizację procedur opisanych w niniejszej normie.

PROGRAM WYKONYWALNY (executable program) — program zawierający wszystkie swoje procedury w postaci nadającej się do wykonania.

REJON KRYTYCZNY (critical region) — część programu sekwencyjnego operująca danymi współdzielonymi — ma ona wyłączny dostęp do tych danych w czasie wykonania.

SEKWENCYJNA KOLEJNOŚĆ OPERACJI (sequential order of operations) — kolejność operacji, których wynik jest taki, jakby były wykonywane ściśle jedna po drugiej.

SEMAFOR (semaphore) — zmienna systemu nadzorczego używana do wymiany informacji synchronizującej między współdziałającymi zadaniami równoległymi. Wszystkie operacje semaforowe tej normy zakładają ochronę za pomocą rejonu krytycznego, zapewnioną przez system nadzorczy.

TRYB DOSTĘPU (access mode) — prawo dostępu lub zezwolenie na dostęp (odczyt lub zapis) do zbioru, udzielone przez procesor w następstwie żądania takiego zezwolenia.

UŚPIONE (dormant) — jeden z określonych stanów zadania. Zadanie uśpione jest znane systemowi nadzorczemu i nie jest w żadnym innym stanie oczekujące, wykonywane lub zawieszone.

WIELOPROGRAMOWANIE (multiprogramming*) — tryb pracy umożliwiający naprzemienne wykonywanie dwóch lub wielu programów komputerowych na pojedynczym procesorze.

WIELOPRZETWARZANIE (multiprocessing*) — tryb pracy polegający na przetwarzaniu równoległym na dwóch lub wielu procesorach.

WIELOZADANIOWOŚĆ (multitasking*) — tryb pracy umożliwiający działanie współbieżne lub naprzemienne wykonywanie dwóch lub wielu zadań.

WYKONANIE (execution) — zespół działań procesora realizującego instrukcje w sposób sekwencyjny.

WYKONANIE POWTARZANE (repetitive execution) — wykonanie polegające na sukcesywnym inicjowaniu zadania bądź w określonych odstępach czasu, bądź przez powtarzające się zdarzenia.

WYKONYWANE (running) — jeden z określonych stanów zadania. Zadanie jest wykonywane w swym procesorze wirtualnym.

ZADANIA RÓWNOLEGŁE, WSPÓŁBIEŻNE (parallel, concurrent tasks) — zbiór zadań, których operacje mogą zachodzić na siebie w czasie.

ZADANIE (task) — obliczenie, które może być zaplanowane do wykonania (ang. scheduled). Operacje tego obliczenia są wykonywane w ściśle sekwencyjnej kolejności.

ZADANIE WSKAZANE (object, designated, referenced task) — zadanie zapoczątkowane, zatrzymywane, zakończone lub poddane innym oddziaływaniom wskutek wywołania procedury systemowej.

ZALEŻNE OD PROCESORA (processor dependent) — działanie nie określone przez niniejszą normę.

ZAWIESZONE (suspended) — jeden z określonych stanów zadania. Zadanie zawieszone jest czasowo zatrzymane i oczekuje na określony warunek, aby kontynuować wykonanie w swym procesorze wirtualnym.

ZDARZENIE (event) — znaczące zjawisko dyskretne lub wypadek, który w planowy sposób wpływa na wykonanie pewnego zadania. Samo zdarzenie pojawia się momentalnie i ustawia znacznik zdarzenia.

ZNACZNIK ZASOBU (resourcemark) — wewnętrzna zmienna systemu nadzorczego używana do wskazania, że zasób jest zarezerwowany wyłącznie dla określonego zadania.

ZNACZNIK ZDARZENIA (eventmark) — wewnętrzna zmienna systemu nadzorczego, używana do wskazania, że zdarzenie zaszło. Jeżeli system użytkowy zawiera zadania równoległe, to znaczniki zdarzeń są dla tych zadań elementami danych współdzielonych.

KAZIMIERZ MALISZEWSKI

■ FUJITSU (Japonia) rozpoczyna produkcję systemów sztucznej inteligencji, składających się z dwóch jednostek LISP-komputerów (tzn. komputerów konstrukcyjnie dostosowanych do wykonywania programów w języku LISP) nazwanych ALPHA. Będzie to najszybsza LISP-maszyna z obecnie produkowanych: 2,5—3 milionów instrukcji na sekundę (MIPS), podczas gdy konkurentki osiągają 0,5—1 MIPS. ALPHA współpracuje ze zwykłym uniwersalnym komputerem, który usprawnia wykorzystanie zasobów sprzętowych. Pamięć tej maszyny składa się z 16 K słów (po 48 bitów) pamięci mikroprogramów, 8 K słów (po 32 bity) stosu sprzętowego przyspieszającego dostęp do danych oraz 4—8 M bajtów pamięci operacyjnej z możliwością wirtualnego adresowania obszarów o rozmiarach do 16 M bajtów. FUJITSU spodziewa się, że ALPHA będzie sprzedawana o połowę taniej od konkurentek.

✷

■ CONTROL DATA coraz bardziej koncentruje się na produkcji pamięci masowych, w szczególności przeznaczonych do stosowania w mikrokomputerach (co oznacza wyroby o szczególnych parametrach technicznych i cenowych). Przewidując rychły zmierzch panującej obecnie technologii Winchester w dziedzinie dysków magnetycznych, CDC intensywnie rozwija nowe konstrukcje — głowice cienkowarstwowe (większa pojemność i zmniejszone koszty produkcji), nośniki metalowe (zamiast tradycyjnych tlenkowych) oraz zapis pionowy. Według pracowników odpowiedzialnych za planowanie i strategię rynkową, za trzy lata 70% pojemności pamięci masowych będzie zrealizowane przez jednostki o średnicy nośnika od 1 do 9 cali, których gęstość zapisu pozwoli osiągnąć 2000 M bajtów w jednym urządzeniu. Obecnie 80% inwestycji służy podjęciu produkcji mikrodysków (o średnicy mniejszej niż 4 cale) i prawdopodobnie większość z zapowiedzianych na bieżący rok 11 nowych modeli będzie mieścić się w tej klasie. W nieco dalszej, ale niezbyt dalekiej przyszłości powinien pojawić się dysk optyczny (pojemność 1 G bajt, cena 4000 dol. za jednostkę).

✷

■ Japoński program rozwoju nowej architektury komputerów i oprogramowania realizującego funkcje sztucznej inteligencji znany pod hasłem p i ą t e j g e n e r a c j i, przynosi już pierwsze efekty. Jak poinformował profesor Tohru Moto-oka z Uniwersytetu Tokijskiego, przewodniczący głównego komitetu koordynującego realizację tego przedsięwzięcia, w czerwcu ub.r. powstał Instytut Nowej Generacji Techniki Komputerowej wraz z laboratorium, którego personel (przede wszystkim projektanci oprogramowania) został oddelegowany z ośmiu japońskich firm komputerowych i będzie opłacany przez trzy lata z funduszy rządowych. Obecnie ukończono prace projektowe nad tzw. maszyną interfejsu (jednym z głównych celów piątej generacji jest stworzenie naturalnego interfejsu człowiek-maszyna), którą do końca roku zrealizuje się przy użyciu konwencjonalnych środków. O ile wyniki badań prototypu będą pomyślne, wkrótce zastąpi ją wersja w postaci układu scalonego VLSI. Jednocześnie kończone są prace, prowadzone na bazie języka PROLOG, nad maszyną relacyjnych baz danych, której fizyczną realizację przewiduje się w roku przyszłym.

✷

■ Firma DIGITAL RESEARCH postanowiła wydawać katalog, który ma zapewnić użytkownikom informacje o licznych i różnorodnych programach pracujących pod nadzorem systemu CP/M, a dostarczanych przez niezależnych sprzedawców. Publikację podzielono na trzy części. W pierwszej — przedstawiono programy według ich producentów (z podziałem na krajowych i międzynarodowych), druga — grupuje programy według ich zgodności z językami opracowanymi przez DIGITAL RESEARCH (wraz z oddzielnym indeksem języków), a trzecia — klasyfikuje producentów zgodnie z dziedzinami, w których się oni specjalizują. Cena katalogu: 10 dolarów.

✷

■ Ciągła presja rynku na producentów układów półprzewodnikowych wywołuje kolejne porozumienia pomiędzy dotychczas konkurującymi firmami. Najczęściej jednak takie umowy dotyczą współpracy przy tworzeniu pewnych konkretnych układów i są ograniczone w czasie. Takie właśnie porozumienie zawarły firmy MOTOROLA i INTEL w celu wspólnego opracowania dwóch modeli pamięci domenowych o pojemności 1 M bitu. Nowe układy mają być unowocześnioną, a przede wszystkim — zmniejszoną, wersją znanego już INTEL 7110. Pierwszy z nich zachowa parametry poprzednika, zaś drugi będzie miał dwukrotnie krótszy czas dostępu oraz dwukrotnie większą prędkość transmisji. Jednocześnie zostanie opracowany nowy model obudowy dla tych układów.

✷

■ Studium wykonane przez TEXAS INSTRUMENTS EUROPE (Francja) zapowiada ośmiokrotny wzrost tegorocznej sprzedaży domowych komputerów (w cenie poniżej 500 dol.) w porównaniu z rokiem 1981. Zgodnie z danymi tej firmy, w roku 1981 sprzedano w zachodniej Europie 300 tys. tych urządzeń, zaś w bieżącym roku oczekuje się osiągnięcia 2,4 mln sztuk (w USA odpowiednio 0,5 i 5,7 mln). Prognozy rynku europejskiego na przyszły rok zapowiadają dalszy 250% wzrost — do ok. 6 mln sztuk. Z badań TEXAS INSTRUMENTS wynika, że 17% nabywców tego typu sprzętu ma mniej niż 16 lat, 19% — od 17 do 25 i 38% — pomiędzy 25 a 40. Raport ten powstał w związku z rozpoczęciem sprzedaży w Europie nowego TI COMPACT COMPUTER 40.

✷

■ W czerwcu br. SHARP CORP. (Japonia) rozpoczął produkcję wyświetlaczy ciekłokrystalicznych, których pojemność jest zbliżona do połowy konwencjonalnego ekranu kineskopowego. Na ekranie mieści się 16 wierszy po 80 znaków lub — w trybie graficznym — 480×128 punktów. Pozostałe parametry: wymiary obrazu — 230×61 mm, głębokość — 18 mm, pobór mocy — 200 mW, waga — 320 g. Użycie układów CMOS z kompensacją temperaturową pozwala zachować wysoki kontrast obrazu w zakresie 0—50°C, przy szerokim kącie widzenia. Te cechy predestynują nowy wyświetlacz do stosowania w komputerach osobistych, przenośnych terminalach, sprzęcie serwisowym czy wreszcie — w motoryzacji. Rzecznik prasowy firmy zapowiedział, że już wkrótce ukaże się następny model o pojemności pełnego konwencjonalnego ekranu.

✷

■ Jedną z podstawowych cech strategii firmy APPLE jest wspieranie małych, niezależnych producentów oprogramowania, nawet pojedynczych osób, tworzących programy dla jej komputerów. Ostatnio, w związku z wprowadzeniem na rynek nowego modelu LISA, APPLE przygotowuje specjalny pakiet ułatwiający produkcję programów. Punktem wyjścia do prac nad nim były języki SMALLTALK i FLAVORS, preferowane przez tych, którzy próbują modelować procesy myślowe człowieka. Natomiast napisano go we własnym firmowym języku CLASCAL, szeroko wykorzystującym koncepcję abstrakcyjnych obiektów zrealizowanych w postaci klas (pojęcie to jest nieobce Czytelnikom INFORMATYKI, gdyż wywodzi się z SIMULI i LOGLANU). Właśnie dzięki klasom programiści spoza firmy będą mogli uzyskać łatwy dostęp do mechanizmów tworzenia obrazu, zwielokrotnienia ekranów czy sterowania myszką, bez potrzeby samodzielnego pisania programów do tych celów. Oczywiście zamiarem takiego działania jest wzbogacenie oprogramowania LISY, rozszerzenie zastosowań, a co za tym idzie — zwiększenie sprzedaży.

✷

■ Laboratorium Komunikacyjne Husashino z NIPPON TELEGRAPH AND TELEPHON PUBLIC CORP. podjęło produkcję pamięci półprzewodnikowej z efektem Josephsona. Pamięć ta, o pojemności 1 K, charakteryzuje się czasem dostępu na poziomie szybkich układów krzemowych (3,3 ns) oraz radykalnie zmniejszonym poborem mocy (2 mW). Co więcej — uzyskano te parametry projektując ją w siatce 5 μm, zamiast 2 μm, która wymaga stosowania najnowocześniejszej, a więc najdroższej techniki litograficznej.

✷

■ Popularność systemu CP/M przekroczyła granice świata mikrokomputerów. W USA rozpoczęto sprzedaż dodatkowych płytek, które pozwalają użytkownikom maszyn PDP-11 korzystać z bogactwa programów napisanych pod kontrolą tego systemu. Płytki te, to w zasadzie jednostki centralne wyposażone w 1—3 procesorów Z80 i 64—192 K bajtów pamięci operacyjnej. Wersja najbardziej rozwinięta dopuszcza wieloprogramową pracę trzech użytkowników jednocześnie.

Majewski J. T., Wagner K.: Język SIMBOL dla MERY 400

INFORMATYKA 1983, nr 10, s. 2

Charakterystyka języka problemowego do przetwarzania danych, dostosowanego do możliwości sprzętowych systemu minikomputerowego MERA 400. Omówiono budowę programu w tym języku, postać danych, wyrażenia, rodzaje instrukcji i zbiorów, obsługę błędów oraz oprogramowanie towarzyszące.

Маевски Й. Т., Вагнер К.: Язык SIMBOL для мини-ЭВМ MERA 400

INFORMATYKA 1983, № 10, стр. 2

Характеристика проблемного языка для обработки данных, приспособленного к аппаратным возможностям системы мини-ЭВМ MERA 400. Обсуждаются структура программы на этом языке, вид данных, выражения, виды инструкций и наборов данных, обслуживание ошибок и сопровождающее программное обеспечение.

Jezierska-Ziemkiewicz E., Ryżko J.: Programowane systemy pomiarowe

INFORMATYKA 1983, nr 10, s. 5

Charakterystyka problemów związanych z postępującą komputeryzacją systemów pomiarowych opartych na technice mikroprocesorowej. Omówiono międzynarodową normalizację magistrali łączącej urządzenia skomputeryzowanego systemu pomiarowego oraz prace krajowe umożliwiające stosowanie takiej magistrali.

Езерска-Земкевич Э., Рыжко Й.: Программированные измерительные системы

INFORMATYKA 1983, № 10, стр. 5

Характеристика проблем, связанных с возрастающим применением автоматизации обработки информации в измерительных системах на базе микропроцессорной техники. Обсуждается международная стандартизация шины, соединяющей устройства измерительной системы, применяющей автоматизацию обработки информации и работы в стране, позволяющие использовать такую шину.

Zalewski J.: Zastosowanie mikroprocesorów 16-bitowych w przetwarzaniu rozłożonym

INFORMATYKA 1983, nr 10, s. 8

Uzupełnienie artykułów zawierających przegląd najpopularniejszych mikroprocesorów 16-bitowych. Zaprezentowano możliwości wykorzystania mikroprocesorów typu Z-8000, MOTOROLA 68000, iAPX 186 oraz NATIONAL 16000 do budowy systemów rozłożonych przestrzennie.

Залевски Й.: Применение 16-битовых микропроцессоров в дистрибутивной обработке

INFORMATYKA 1983, № 10, стр. 8

Дополнение статей, содержащих обзор наиболее популярных 16-битовых микропроцессоров. Представлены возможности использования микропроцессоров типа Z-8000, MOTOROLA 68000, iAPX 186 и NATIONAL 16000 для конструирования систем дистрибутивной обработки.

Przemysłowy FORTRAN czasu rzeczywistego. Część 1

INFORMATYKA 1983, nr 10, s. 11

Omówienie projektu międzynarodowej normy rozszerzającej język FORTRAN dla zastosowań do sterowania procesów przemysłowych. Omówiono dotychczasowy rozwój prac normalizacyjnych oraz część projektu obejmującą definicję procesów działania w czasie rzeczywistym.

Промышленный FORTRAN реального времени. Часть 1

INFORMATYKA 1983, № 10, стр. 11

Обсуждение проекта международного стандарта, расширяющего язык FORTRAN на применения в управлении промышленными процессами. Обсуждается ход выполненных до сих пор работ по стандарту и часть проекта, охватывающая определение процессов действия в реальном времени.

Wilczek T.: Aspekty użytkowe języków symulacyjnych (2). Symulacja ciągła i ciągło-dyskretna

INFORMATYKA 1983, nr 10, s. 15

Omówienie problemów programowania symulacji ciągłej oraz ciągło-dyskretnej na tle charakterystyki najczęściej stosowanych do tych celów specjalizowanych języków programowania.

Вильчек Т.: Пользовательские аспекты языков моделирования (2). Непрерывное и непрерывно-дискретное моделирование

INFORMATYKA 1983, № 10, стр. 15

Обсуждение проблем программирования непрерывного и непрерывно-дискретного моделирования на почве характеристики применяемых чаще всего для этих целей специализированных языков программирования.

Ryznar Z.: Pakiet dokumentowania i wspomagania prac programistycznych PROSTR

INFORMATYKA 1983, nr 10, s. 18

Charakterystyka pakietu programowego PROSTR, opracowanego przez ZETO Kraków z przeznaczeniem do stosowania na komputerach Jednolitego Systemu z wykorzystaniem systemu operacyjnego OS. Omówiono konstrukcję pakietu oraz sposób i warunki jego wykorzystywania.

Рызнар З.: Пакет документирования и поддержки работ в области программирования — PROSTR

INFORMATYKA 1983, № 10, стр. 18

Характеристика программного пакета PROSTR, разработанного ZETO Краков, предназначенного для применения на ЕС ЭВМ под управлением операционной системы ОС ЕС. Обсуждена конструкция пакета, а также способ и условия его использования.

Majewski J. T., Wagner K.: SIMBOL — language for MERA 400

INFORMATYKA 1983, No. 10, p. 2

Problem oriented language for data processing applications on MERA 400 minicomputer system is characterized. Program's structure, data formats, expressions, statements and files, error handling and supplementary software are discussed.

Majewski J. T., Wagner K.: SIMBOL-Programmiersprache für MERA 400

INFORMATYKA 1983, Nr. 10, S. 2

Eine Charakteristik von Problemsprache für Datenverarbeitungsanwendungen, die zur Hardwaremöglichkeiten des MERA 400 Kleinrechnersystems angepasst ist. Es wurden die Programmstruktur, Datengestalt, Ausdrücke, Anweisungs- und Dateisarten, Fehlerbedienung und Ergänzungssoftware besprochen.

Jezierska-Ziemkiewicz E., Ryżko J.: Programmed measurement systems

INFORMATYKA 1983, No. 10, p. 5

Problems connected with computerization of measurement systems based on microprocessor technology are characterized. International standarization of the bus for devices connection in computerized measurement system, as well as home research for creating possibility of the bus application are discussed.

Jezierska-Ziemkiewicz E., Ryżko J.: Programmierte Messsysteme

INFORMATYKA 1983, Nr. 10, S. 5

Eine Charakteristik von Problemen, die mit fortschreitenden Computerisierung der Messsysteme auf Basis von Mikroprozessortechnik verbunden sind. Es wurden internationale Normierung der Sammelschiene für Verbindung eines computerisierten Messsystems und einheimische Forschungsarbeiten, die Anwendung einer solchen Sammelschiene ermöglichen, besprochen.

Zalewski J.: 16-bit microprocessor application in distributed processing

INFORMATYKA 1983, No. 10, p. 8

Appendix for articles with most popular 16-bit microprocessor surway. Possibility of Z-8000, MOTOROLA 68000, iAPX 186 and NATIONAL 16000 microprocessors application for building distributed systems is presented.

Zalewski J.: Anwendung von 16-Bits-Mikroprozessoren in verteilter Datenverarbeitung

INFORMATYKA 1983, Nr. 10, S. 8

Eine Ergänzung von Artikeln mit Vorschau der populärsten 16-Bits-Mikroprozessoren. Es wurden Anwendungsmöglichkeiten von Z-8000, MOTOROLA 68000, iAPX 186 und NATIONAL 16000 Mikroprozessoren in verteilten Systemen angegeben.

Real-time FORTRAN for industrial applications. Part 1

INFORMATYKA 1983, No. 10, p. 11

Proposal of international standard referring to FORTRAN extension for applications of technologic process control. Development of standarization work and proposal's part containing definition of real-time processes are discussed.

Echtzeit FORTRAN für industrielle Anwendungen. Teil 1

INFORMATYKA 1983, Nr. 10, S. 11

Eine Besprechung des internationalen Normenentwurfes, der eine FORTRAN-Erweiterung für Steuerung der technologischen Prozesse umfasst. Es wurden bisherige Normierungsarbeiten und der Entwurfsteil mit Definierung von Echtzeitprozessen besprochen.

Wilczek T.: Utility aspects of simulation languages (2). Continuous and continuous-discrete simulation

INFORMATYKA 1983, No. 10, p. 15

Programming problems of continuous and continuous-discrete simulation and characteristics of most often used for this purpose programming languages are discussed.

Wilczek T.: Nutzaspekte der Simulationssprachen (2). Fortlaufende und fortlaufend-diskrete Simulation

INFORMATPKA 1983, Nr. 10, S. 15

Eine Besprechung von Programmierungsproblemen der fortlaufenden und fortlaufend-diskreten Simulation mit einer Charakteristik der am meisten für diese Zwecke angewendeten Programmiersprachen.

Ryznar Z.: PROSTR — a package for programmer's work documenting and assisting

INFORMATYKA 1983, No. 10, p. 18

Characteristics of the PROSTR program package, elaborated by ZETO Kraków, for application on Unified System computers using OS operating system. The package's architecture, as well as method and conditions of its application, are discussed.

Ryznar Z.: PROSTR — ein Pakett für Dokumentierung und Unterstützung von Programmierarbeit

INFORMATYKA 1983, Nr. 10, S. 18

Eine Charakteristik des PROSTR-Programmpakets, der von ZETO Kraków für Anwendung auf ESER-Rechnern mit Ausnützung des OS-Betriebssystems erarbeitet wurde. Es wurden die Struktur des Pakets, sowie Technologie und Bedingungen seiner Ausnutzung besprochen.

Komputery zagrażają!

Opisane poniżej fakty miały miejsce w sierpniu br. Informacja ta jest konieczna, może się bowiem zdarzyć, że zanim INFORMATYKA dotrze do Czytelnika, to i owo ulegnie zmianie. Ale nawet wówczas pouczająca — jak sądzę — będzie chwila zadumy nad ukazanymi mechanizmami.

Dobry znajomy, po powrocie z zagranicznych wojaży, gdy już pierwsze wrażenia zostały opowiedziane, zadał słuchającym pytanie: kto zgadnie, jakich rzeczy przywozić do Polski nie wolno (chyba, że za specjalnym, imiennym zezwoleniem Głównego Urzędu Ceł); dla ułatwienia — są takie trzy pozycje (nie licząc: broni, narkotyków, pornografii). Zgadujących było kilku, typy na ogół rozstrzelone, nietrafne. Otóż: po pierwsze — nie wolno przywozić materiałów i urządzeń poligrafii (westchnęliśmy ze zrozumieniem, choć chyba już do śmierci się nie dowiem, dlaczego słowo pisane jest tak ważne — ważniejsze niemal od realnej rzeczywistości); po drugie — materiałów i urządzeń łączności (znowu skwitowaliśmy to gestami zrozumienia); wreszcie po trzecie — i tu dramatyczna przerwa — materiałów i urządzeń do przetwarzania danych.

To nas istotnie zamurowało. Jakże bowiem śmiesznie na tym tle (choć to tragiczny paradoks) brzmią postulaty PTI i KI PAN o subsydiowaniu niektórych zakupów sprzętu informatycznego za granicą, postulatów sformułowanych i przedstawionych naczelnym władzom wiele, wiele miesięcy temu. Natychmiast posypały się hipotezy — co też może być przyczyną zakazu. Wśród wielu ostały się dwie:
— albo komuś pomyliła się informatyka z informacją, a więc znowu — pierwszy zakaz
— albo ktoś gdzieś wymusił.

Wszyscy byli skłonni przyjąć drugi wariant. Mikrokomputery są już przecież na naszym rynku za nieliche pieniądze (dolar wypada za ok. 1000 zł) i widać opłaca się przyhamować (a może w ogóle zlikwidować) próby przywożenia do Polski mikrokomputerów „na dziko". Na koniec wymieniliśmy, po raz nie wiem już który, smutne refleksje o informatyce oraz reformie korzystnej dla monopolistów.

Uciechę dawało mi tylko przekonanie, że przynajmniej mam temat dla INFORMATYKI. Trzeba było jednak informację sprawdzić, a może więcej — usłyszeć zdanie pomysłodawców zakazu. Główny Urząd Ceł mieści się w Warszawie, centrala telefoniczna kompetentnie łączy z wydziałem informującym o przepisach przywozowo-wywozowych. Przedstawiłem się tam jako redaktor INFORMATYKI.

Miałem więc okazję rozmawiać z kilkoma osobami z GUC, wśród nich także rzecznikiem prasowym. Wszyscy rozmówcy byli nad wyraz życzliwi, co mile zaskoczyło, niektórzy pocieszali, że zakaz nie jest groźny, że Urząd — jak do tej pory — nikomu nie odmówił zezwolenia, że nie powinienem się tak przejmować. Jeden z rozmówców na moje natarczywe dopominanie się wyjaśnień, uśmiechając się, rzucił coś w rodzaju: a bo to wiadomo, co też na tych komputerach będą ludzie robić; i zawiesił znacząco głos — że niby on wie, a ja rozumiem, o co chodzi.

Nie dałem za wygraną, coraz bardziej zaciekawiony. Informacja, którą na koniec uzyskałem, przeszła wyobrażenia. Otóż zakaz został wydany przez Ministra Finansów (tak się dzieje zawsze w sprawach celnych — GUC podlega temu ministerstwu) na wniosek... Ministra Przemysłu Maszynowego, z lipca 1981 r. Niesamowite, nigdy by mi się nawet nie przyśniło, że producenci śladowych ilości urządzeń informatyki w Polsce tak ochoczo zadbają o ochronę swoich interesów. I że nikt nie ukróci tej szkody wyrządzonej polskiej informatyce.

Ot i cała historyjka. Zapewne ten i ów wzruszy ramionami — o co tu chodzi, gdzie te wstrząsające fakty, ot jeden jeszcze partykularyzm, jeden jeszcze idiotyczny przepis, a w ogóle czy ludzie tak chętnie przywoziliby te mikrokomputery za zielone, kiedy znacznie lepiej jest przywozić kawę, samochody lub diabli wiedzą co jeszcze.

Upieram się jednak, że jest o czym mówić. Piorunować na całokształt jest wprawdzie zajęciem łatwym, ale równie jałowym. Tu zaś na małym, namacalnym przykładzie widać jasno i niedostatki reformy gospodarczej w Polsce, i brak polityki wobec informatyki (podkreślam — wobec informatyki, a nie monopolistów produkujących sprzęt; w znikomych zresztą ilościach).

Nie raz już i nie dwa, wielcy od reformy zapowiadali jeden z najważniejszych aktów ustawodawczych — ustawę antymonopolistyczną. Jak na razie nic z tego, zjednoczeń wprawdzie nie ma, ale co tam, są zrzeszenia, oczywiście — dobrowolne. Zresztą, czy ktoś kiedyś słyszał o przymusowym monopolu? Przecież dla producenta monopol to „palce lizać", sam się do tego garnie, gdy ma okazję. Odbiorca znów stoi sam wobec możnych producentów, których na dodatek wspierają wysokie kręgi administracji.

A co z polityką wobec informatyki? Ano nie ma. Nie ma we władzach organu, który kreowałby tę politykę. Są różnorakie ministerstwa — jedno, odpowiedzialne za produkcję, wnioskuje obronę tej produkcji przez drastyczną ochronę celną, drugie — przepis wydaje, a nie ma już nikogo, kto zastanowiłby się, jaki to ma sens w obecnej sytuacji polskiej informatyki.

Na świecie mikrokomputer staje się tak powszechny, jak telefon, maszyna do pisania czy telewizor. Są kraje gdzie od studentów wymaga się, aby mieli własne mikrokomputery. Ale nam, oczywiście, na „nowinkach" nie zależy, dla nas to sprawa tak odległa, że wręcz nie warto się tym zajmować, a nawet mówić o tym.

No cóż, mam nadzieję, że powyższe słowa przejdą wreszcie do wstydliwej przeszłości. Nadal nie potrafię pozbyć się optymizmu.

JANUSZ GWIAZDA

Zasady prenumeraty

Zamówienia i przedpłaty na prenumeratę INFORMATYKI przyjmuje Zakład Kolportażu Wydawnictwa NOT SIGMA. Adres pocztowy: Wydawnictwo NOT SIGMA — Zakład Kolportażu, 00-950 Warszawa, skr. poczt. 1004. Konto bankowe: 1036-7490-139-11, III O/M NBP w Warszawie.

Jednostki gospodarki uspołecznionej, instytucje i organizacje przesyłają zamówienia (w 1 egz.) zawierające: tytuł czasopisma, liczbę zamawianych egzemplarzy, okres prenumeraty i pełny adres zamawiającego z kodem pocztowym, a także numer konta bankowego zamawiającego oraz (ewentualnie) adresy odbiorców, którzy na zlecenie zamawiającego mają prasę otrzymywać.

Dopisując w zamówieniu PRENUMERATA STAŁA, zamawiający nie będzie musiał corocznie ponawiać zamówienia, a jedynie dokonywać przedpłaty według aktualnie obowiązujących cen.

Warunkiem realizacji zamówienia jest równoczesne dokonanie odpowiedniej wpłaty na ww. konto Wydawnictwa NOT SIGMA.

Prenumeratorzy indywidualni dokonują wpłaty przekazem na ww. konto, podając na odwrocie odcinka dla adresata-posiadacza rachunku: tytuł czasopisma, liczbę zamawianych egzemplarzy oraz okres prenumeraty.

Zamówienia i wpłaty przyjmowane są w terminach:

- do 15 listopada — na I kwartał, I półrocze i cały rok
- do 28 lutego — na II kwartał
- do 31 maja — na III kwartał i II półrocze
- do 31 sierpnia — na IV kwartał.

Uwaga: przy podawaniu kodu pocztowego i numeru konta bankowego obowiązuje bardzo czytelne pismo.

Prenumerata kwartalna — 225 zł, półroczna — 450 zł, roczna — 900 zł. Prenumerata ze zleceniem wysyłki za granicę jest dwukrotnie droższa.

Od 1 lipca br. wprowadzona została prenumerata ulgowa dla członków stowarzyszeń naukowo-technicznych NOT, studentów wyższych uczelni, uczniów szkół zawodowych. Warunkiem jej uzyskania jest poświadczenie blankietu dla nabywcy indywidualnego (na odcinku dla adresata) przez właściwe stowarzyszenie NOT, wyższą uczelnię lub szkołę zawodową. Cena prenumeraty ulgowej: kwartalnie — 150 zł, półrocznie — 300 zł, rocznie — 600 zł.

Dodatkowych informacji o prenumeracie udziela: Zakład Kolportażu, tel. 40-00-21 w. 293, 295, 217 oraz 40-35-89. Egzemplarze archiwalne można nabywać w Klubie Prasy i Informacji Technicznej w Warszawie, ul. Mazowiecka 12, tel. 27-43-65.

W NUMERZE 11/83 PRZEDSTAWIMY:

Błaszczak S., Iszkowski W.: Język ADA w testach

Grabowski J.: Porównanie właściwości mikroprocesorów 16-bitowych. Część 1

Abramowicz A.W.: System automatycznego redagowania tekstów literackich

Przemysłowy FORTRAN czasu rzeczywistego. Część 2

Warski W.: Alternatywa dla struktury blokowej w językach czasu rzeczywistego

Stepaniec J., Stasiewicz P.: Automatyzacja obsługi użytkownika zdalnego w systemie operacyjnym GEORGE-3

W NAJBLIŻSZYCH NUMERACH:

- Przegląd języków wysokiego poziomu
- Przetwarzanie strumieniowe
- iAPX 432
- Komputerowe wspomaganie prac inżynierskich